University of Southern Queensland
Faculty of Engineering and Surveying

Remote Monitoring and Instrumentation for Mount Kent Observatory

A dissertation submitted by

Willy ONG

in fulfillment of the requirements of

Course ENG4111 and ENG 4112 Research Project

towards the Degree of

Bachelor of Engineering (Software Engineering)

Submitted: October, 2005

Abstracts

This thesis explores the development of web interface design for Mount Kent Observatory. It also investigates remote observing, and planetary photography using webcam.

USQ maintained an observatory at Mount Kent, 30 km Southwest of Toowomba. The observatory is undergoing development process with purpose of providing remote and robotic observing for distance education students. With longitude difference, student in daytime classes in United States will have the opportunity of live observing the night sky above Australia. The purposes of the web interface design are to provide weather information for Mount Kent to assist remote observing, to implement live observing, and to include an image gallery. This project examines the software that provides web-base access to remote robotic observing, and research planetary photography using webcam. Webcam is superior to CCD camera in non-deep sky imaging. With image stacking, planetary images captured by webcam can be processed further to generate rich image. The implementation of live webcam preview in web interface design can also be use in future live observing.

Future work in implementing the weather station and cloud sensor output into the web interface design may need to be considered. The web interface design may need further development and maintenance.

University of Southern Queensland

Faculty of Engineering and Surveying

Eng4111 & ENG 4112 *Research Project*

Limitation of Use

Certification

I certify that the ideas, designs and experimental works, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Willy Ong

Student Number: 0031134295

Signature

Date

Acknowledgement

The completion of this project would not have been possible without the assistance, guidance, support and encouragement of my supervisors, my parents and those who involved in Mt Kent Observatory development.

I am grateful to my supervisors: A. Prof. John Leis and Dr. Brad Carter for their suggestion on taking and making proposal for this project; and their guidance, comment, and help in finishing this project and dissertation. Thanks most also go to Dr. Rhodes Hart for his support in acquiring information related to the project.

Finally, I would like to express my gratitude to my parents who have been very caring, understanding, and loving. Without their support, my study in Australia would never been possible.

Willy ONG

University of Southern Queensland

October, 2005

Table of Content

List of Figures

List of Tables

Chapter 1

Introduction

USQ maintains an observatory at Mount Kent, 30 km southwest of the Toowoomba campus in the Darling Downs region of Southern Queensland. The primary purpose of Mount Kent Observatory (MKO) is to provide remote and robotic observing for distance education students, including those enrolling in a new Graduate Certificate in Astronomy to be offered by USQ in 2005. The observatory originally has three domes (Mt Kent Observatory 2004), each housing a telescope including:

1. Webb Telescope: A 40cm aperture f/10 Meade Instruments computer-controlled LX200 Schmidt-Cassegrain telescope. The instrument is used for student and public viewing nights.

2. O'Mara Telescope: This new telescope comprises a Paramount ME robotic telescope mount equipped with a Celestron-14 35cm aperture and a CCD camera. This instrument is used to support staff and student research.

3. Tamborine Telescope: This telescope is expected to be replaced by a new 50cm telescope in 2005.

In addition to the telescopes, the Mt Kent site houses the Educational Development Group (EDG) building with a lecture room, accommodation facilities and a control room for indoor operation of the observatory telescopes. A weather station is also located on the Observatory grounds. A fast communications link and equipment including a microwave tower has been installed to suit remote observing, and a weather station.

Figure 1.1: Mount Kent Observatory. Webb Telescope (front dome) O'mara Telescope (middle dome), Tamborine Telescope (back dome). (Adapted from: Carter (2005)).

In 2004, A Digital Science Partnership Project was proposed by University of Louisville, Kentucky, USA. This Project involves developing the capability to remotely and robotically operate MKO observing facility, for teaching, outreach, and research purposes. With a longitude difference that enables the night sky above Australia to be observed live by students in daytime classes in United States, the Southern latitude of the site offers a chance to view and carefully study the rich skies around the Milky Way centre, the Magellanic Clouds, and even recent and pending supernovae otherwise inaccessible to US observers. (Department of Physics, 2004).

The Digital Science Partnership Project involves installation and operation of a Celestron-20 at Mt Kent Observatory. To accommodate this, The Page dome will be refurbished to house the new telescope with cameras, computers, and communications equipment for both live remote-control and automated observing.

In addition to the Digital Science Partnership project, a substantial research interest has been generated in the 'remote observing' project, which aims to have an Internet-viewable and controllable telescope available to users anywhere in the world. The installation and development of remote robotic control for O'Mara Telescope is carried out by Dr. Brad Carter and Dr. Rhodes Hart. With regard to this issue, this topic is suggested by Dr. John Leis and Dr. Brad Carter. This topic was then proposed and chosen for the research project.

1.1 Aim and Objectives

The project aims to create a web interface design that provides remote observation and instrumentation for the USQ-maintained observatory at Mt Kent, by presenting recent or current telescope camera images, weather information for Mt Kent, weather station output, and if possible remote control of the telescope. Specifically, the objectives of the project are:

1. Design a web interface which encapsulates:
   - Recent satellite imagery, lightning strike information, and rain radar with Mt Kent pinpointed
   - Telescope (CCD) camera image and image gallery

   The web interface design objectives take in the study of web design.

2. Design an Internet–viewable display of recent telescopic images captured. This involves preliminary research information related to software which controls the

Mt Kent Observatory instruments, including the telescope, camera, and the dome itself; and research the communications infrastructure available to Mt Kent Observatory and USQ, including the Apache web server for the observatory.

As time permits,

3. Inclusion of weather station output and cloud sensors in the web interface design.
   The weather station output and cloud sensors are expected to be finished earlier so that these can be integrated into the web interface design.

4. Provide remote access to software that controls the telescope by web interface.
   Without support of third party software which interfaces with software that control telescope instruments, remote observing would never be accomplished within the time constraint as it requires abundant work.

5. Undertake a security audit of the Mt. Kent technology infrastructure
   The security aspect is of paramount importance, especially when remote access to the observatory controls is available. In the project, following security systems will be considered:
   - User account and login security
   - User access permission level
   - Firewall and network connection
   - Operating system and system backup

1.2 Scope and Constraint

The scope and constraint is derived from project specification. Since the final deliverables of the project is a web interface design, this section will focus more on web interface design. Following is the summary:

- The project activity does not include the installation and configuration of the remote robotic telescope. This task required more knowledge, time and access authority to be accomplished.

- The web interface design shall include:
    - presentation of recent Rain Radar Map, Satellite Imagery, and Lightning trackers with Mt Kent pinpointed;
    - Internet-viewable recent telescopic image;
    - Image gallery;
    - Weather station and cloud sensor output (optional);
    - Remote access to observatory telescope (optional); and
    - User access and login security.

- The web interface design shall be supportable by globally used web browsers.

- The layout for web interface design shall be simple and easy to read. The used of colour, background and graphic shall not distract user from the main context of the webpage.

Chapter 2

Design Methodology

With regards to project aim and objectives, the project task can be decomposed further into smaller activities. In brief, the project consists of following activities:

1. Research software which controls O'Mara telescope equipment;
2. Research the communications infrastructure available to MKO;
3. Research planetary photography by using webcam;
4. Research CCD image capture and processing software;
5. Research web design method;
6. Web interface design;
7. Weather station output and cloud sensors in the web interface design;
8. Research remote access to software that controls the telescope;
9. Undertake a security audit of the MKO technology infrastructure; and
10. Verification and validation of the web-interface design

This chapter will present activities that had been carried out in project development. Webcam research was added later during project development as there is an emerging interest in non-deep sky photography using webcam. Some of the latter few activities were not finished completely as there were few changes during project development.

2.1 Telescope Equipment and Software

In obtaining information regarding the telescope, a number of astro-meetings had been attended during the first semester, and a few field trips to Mt Kent were carried out. The dome which housing O’Mara Telescope has following equipments installed:

- Telescope: Celestron CGE1400, 14” aperture
- Focuser: OPTEC Celestron 3”
- Mount: Paramount ME Robotic Telescope Mount
- Camera: SBIG New Large Format camera

The software that controls these equipments was installed into a personal computer (PC) which runs on Windows OS. The reason why Windows was chosen is because the main software works only on Windows platform. Dr. Brad Carter informed in early astro-meeting that the software which controls the equipments is:

- *The SKY*: controls the telescope movement (mounting) for tracking sky objects. (SoftwareBisque 2005b)
- *Maxim DL*: controls CCD camera, filter wheel, focuser for CCD imaging, also integrated with image processing tools (Diffraction Limited, 2004)
- *Automadome*: controls robotic domes to keep the dome's slit synchronized with the telescope's optic (SoftwareBisque 2005a).
- *Astronomy Control Panel (ACP)*: provides interface for software above, also featuring image stacking, Web browser and FTP (remote) access, etc. (Denny 2005a)

The installation of this software and equipment has been carried out by Dr Brad Carter and Dr Rhodes Hart. The interface between the software was working during testing of *The SKY* in tracking sky object. A few problems found later, regarding the image capturing by SBIG Camera. The latest problem was related to inaccuracy in tracking sky object. Due to the circumstances and limitation, the research for the project can only be done without having the instruments fully working. Apart from the installation and configuration progress, the main issue is the license of the interface software, *ACP* that provides remote imaging, was already purchased. This concluded that the research on remote access to telescope will be conducted on ACP rather than software selection or software design. Moreover, the importance of research CCD image capture software and processing software is insignificant as this features is included in *ACP*

2.2 Communication Infrastructure

The main communication equipment is the microwave disc which links Mt Kent Observatory to USQ Toowoomba Campus. The data transfer rate is 4 Mbps and this is fast enough for networking. At MKO site visit, it was found that the PC, which housed by O'Mara Telescope, already has internet access and so on with those inside EDG Building. This summarises that MKO is ready for internet acces and no further communication infrastructure research is needed.

2.3 Webcam and Image Processing Software

Telescope imaging using webcam is popular nowadays for capturing image of bright objects such as moon and planets. This is because using telescopic camera to observe a large, bright object such as a planet is much more affected by atmospheric turbulence than to observe dim deep-sky objects. In the contrary, a webcam turn out to be a perfect device for planetary photography as the atmospheric turbulence does not have much affect compared with telescopic CCD camera. Webcam takes a number of frames (i.e. photo) in a row to produce a video. A good planetary photography can then be created by accumulating the photos (frames), deciding the good ones, and stacking them together (Kurkowski 2005). Many astronomers who use webcam for planetary imaging found that webcams are superior to CCD camera. However, not all webcam are suitable for planetary photography. The only suitable webcam is the one with a real CCD sensor (which usually found in astrophotography camera). Following are the most popular CCD webcam for astrophotography:

Table 2-1: Webcan for Planetary Photography

Model	Video capture resolution(pixel)	Video capture speed (Frames per second)
Philips ToUcam ProII	640 x 480	60
Philips ToUcam Pro	640 x 480	60
Philip Vesta Pro	640 x 480	30
Philip Vesta	640 x 480	30
Logitech Quickcam Pro 4000	640 x 480	30
3Com Home Connect	640 x 480	60

Some of the best freeware for processing Webcam image are: K3CCD Tools, RegiStax, and IRIS. The software chosen to be used in project activity is RegiStax because of its popularity and automatic feature. The implementation and testing will be discussed later in Chapter 5.

2.4 Web Design Method

The study of Website design went along with the design procedures itself. This activity is the most time consuming as there was no experience or previous study on web interface design. This is because the Web Publishing course, which offered by the university was not taken, as it was an elective course for Software Engineering Degree Program. This project was chosen as it is challenging to learn new programming language. To start with, a copy of Web Publishing study book was purchased from USQ Bookshop for self-study. The remaining of the study was through website and online forums.

The website design method that was studied and implemented are: HyperText MarkUp Languange (HTML), Cascading Style Sheet (CSS), and JavaScript. HTML and CSS are the foundation of website design. JavaScript supports embedded programs which are executed by the client's (user) browser. PHP is a server-side HTML embedded scripting language. It provides web developers with a full suite of tools for building dynamic websites and support server-side development.

PHP excels on the server-side but has to setup elaborate two-state processing. PHP is preferred if extensive client side validations and coding are required. JavaScript on the other hand easily handles the client side validations and other operations. Also because web browsers offer more uniform CSS and DOM support, it has become a much more viable web development platform (The Open Source 2004). Since the web interface design does not need that much server function. The study of PHP scripting is therefore excluded. The web design concept and implementation will be discussed on next Chapter

2.5 Weather Station and Cloud Sensor

Weather station and cloud sensor output are to be displayed in the website design for benefit of remote operation of telescope. These equipments are expected to be installed earlier at Mount Kent Observatory (the installation is not part of the project). However, due to time limit, these equipments are not ready for use yet. Dr. Rhodes

Hart had recently obtained a cloud sensor. In conclusion, the weather station and cloud sensors output are not included in the project development.

2.6 Security Audit

Security is the most important if remote access to telescope control is validated as it exposes the infrastructure more to internet threat such as viruses and hackers attack. The consequences may include hardware damages (dome, telescope equipment, camera, and PC), system damages (PC's software and network system), and Confidential Information leakage. Thus a security audit of the Mt. Kent technology infrastructure is necessary including: User account and login security, user access permission level, firewall and network connection, operating system and backup.

ACP provides web server and FTP server to registered user. User account, access and login are protected by password system. ACP security system is quite promising. On the other hand, the internet connection is through USQ network. USQ network system has promising firewall and virus protection. Since, the remote access is currently not established yet, it can be conclude that currently the system is quite secure from internet threat.

Chapter 3

Web Interface Design

Web interface Design is the longest activity in this project as it is the final deliverable. The entire development process was done under Windows XP platform. The tools that were use in development and debugging the website are: WordPad, Internet explorer 6, and Mozilla 1.7.12. The only debugging tool that was used is JavaScript Console provided by Mozilla web browser.

The main reason for using WordPad rather than the Mozilla composer is because the style of automatic indentation of the composer is not preferred. Both Internet Explorer and Mozilla are quite stable and popular web browser. Internet Explorer comes with Windows Operating System (OS). Mozilla is a powerful open source browser based on gecko platform and available for most Operating System. Web browsers are very important information for website development because the presentation or execution of a web interface design will be different when using different web browser. Thus, in the project, the web interface design should be supportable by both of these popular browsers.

This chapter is concerned with the process, concept and algorithm used in developing the website in the fulfilment of the project.

3.1 Design Architecture

The architecture of the web interface design was deduced from project specification. The web pages for the website design are:

1. *index.html*

   This is index or default page for Mt Kent Observatory. It contains a brief introduction to Mt Kent Observatory and presents recent weather information. The weather maps to be covered are: rain radar map, satellite imagery, and lightning tracker adapted for Mt Kent Observatory.

2. *telescope.html*

   Since O'Mara telescope is not ready for remote robotic control yet, this page is build for its future remote observing project. A webcam live preview is also implemented in this page for planetary observation. The input hardware for live display can be any input such as CCD camera, camcorder.

3. *gallery. Html*

   This page will be used to publicise image captured by MKO's telescopes.

4. *temp.html*

   This page is a temporary page for *weatherstation.html* that presents the output of recent weather station and cloud sensor. The weather equipments are going to be installed at MKO.

There is not much web pages needed to be prepared for Mount Kent Observatory at this stage. Thus one level of navigation is enough, that is, all of the web pages have same access level, linked to each other through navigation list. Additional pages are added as they are basically included or required in web interface design, these are: *contactus.html* and *Disclaimer.html*.

3.2 Weather Information

The Weather maps that are needed to be presented in the website design are: satellite imagery, lightning strike information, and rain radar map. Extensive search in internet was performed to find the best maps for Mount Kent Observatory. The best weather imagery found in the progress is:

- *Satellite imagery*: Coloured Infrared Satellite image provided by Bureau of Meteorology, Australia.
- *Rain Radar Map*: Marburg Rain Radar Map provided by Bureau of Meteorology, Australia; and
- *Lightning strike information*: Lightning Tracker provided by Energex.

3.2.1 Satellite Imagery

The coloured infrared Satellite image provided by Bureau of meteorology Australia can be downloaded freely from their web server and FTP server. The link is as follow:

- Website: http://www.bom.gov.au/gms/IDE00035.jpg
- FTP: ftp://ftp2.bom.gov.au/anon/gen/gms/

This image is originally captured by Japan Meteorological Agency geostationary satellite MTSAT-R and processed by the Bureau of Meteorology. The Satellite image is taken periodically at interval of one hour. The time stamp on the images is the start time of the reception of the top of the image from the satellite. It takes approximately 30 minutes to complete a scan and the image is generally available on the Web within 30 minutes of completion. Thus, the minimum delay in satellite image publication is about one hour and the maximum is 2 hours (Bureau of Meteorology Australia 2005c).

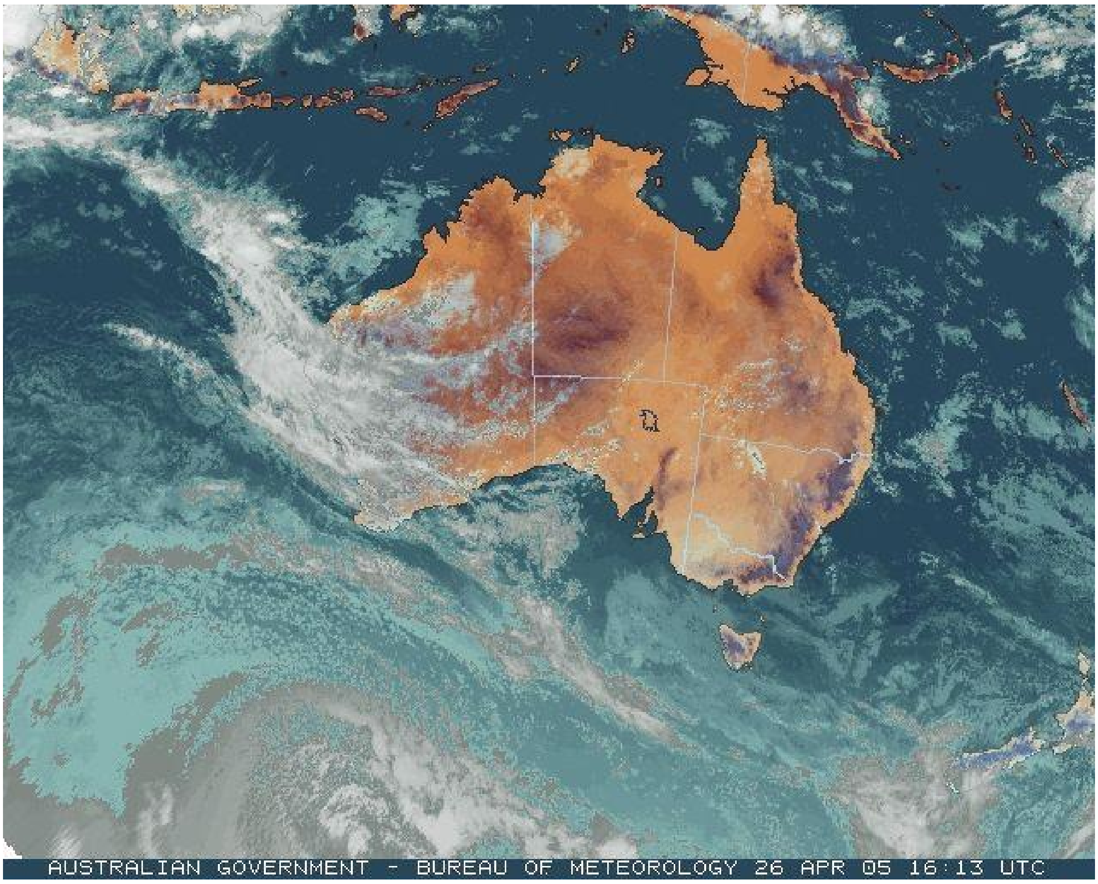

Figure 3.1: Coloured IR Satellite Image (Adapted from Bureau of Meteorology Australia (2005c))

The FTP server contains a collection of recent coloured infrared satellite image. The website design includes animated loop display by acquiring a series of four images and manipulating the visibility. The satellite images stored in the FTP server are named in timestamp. In order to acquire these image files, JavaScript programming and testing were done to bring out fully working code as below:

```
function getLatestImage(target) {
    // get current datetime and create datetime with lag
    var now = new Date();
    milli = Date.parse(now.toString());
    lag = (1 + count)*3600000;          // extra one hour lag for publication delay
    milli -=lag;
    var nowLag = new Date(milli);

    // get specific string format for datetime
    var years= nowLag.getUTCFullYear();
    var months= nowLag.getUTCMonth() + 1;  // the month range from 0-11, thus add 1
    var dates= nowLag.getUTCDate();
    var hours= nowLag.getUTCHours();
    dd = mm = hh = "";
    if (months < 10) {                    // convert to two digit
        mm = '0';
```

```
    }
    if (dates < 10) {                       // convert to two digit
        dd = '0';
    }
    if (hours < 10) {                       // convert to two digit
        hh = '0';
    }

    // concat the datetime string format and create pathname
    timeValue = ''+ years + mm + months + dd + dates + hh + hours + '30';
    pathname = 'ftp://ftp2.bom.gov.au/anon/gen/gms/IDE00035.'+timeValue+'.jpg';
    // assign pathname to image source
    target.src = pathname;
    count++;
}
```

The code above can be explained in sequence as following:

- *Get current date and time*
  This is done by creating a new `datetime` data object

- *Deduct it by number of hours lag*
  The `datetime` needs to be converted to amount of time, then the amount of time lag needs to be deducted from it. This is done in millimetre as the unit used to store `datetime` data format is millimetre.

- *Convert the `datetime` to specific format as those filename in FTP server*
  The satellite image files are stored in timestamp format. Following is the pathname of one of the file stored in BOM's FTP server:
  ftp://ftp.bom.gov.au/anon/gen/gms/IDE00035.200510191030.jpg
  The last part of the address is in date and time format, that is 200510191030 which equal to year (4 digit), month (2 digit), date (2 digit), hour (2 digit), and minute. The minute is the same for all satellite images: 30 and so with the remaining pathname. Thus, to get the desired file, the current or lagged `datetime` value must be converted into this format. The month value is added with one as the JavaScript `getUTCMonth()` method return 0 to 11 instead of 1 to 12.

- *Create pathname*
  The desire pathname can be created by concatenating the specific timestamp format with remaining pathname.

- *Assigned the pathname to image source*
  The pathname is then assigned to the image source of IMG element of HTML document

- *Increase lag counter for next image acquiring*
  The time lag is needed to be increased for acquiring next lagging image

In the development, an online JavaScript reference (Refsnes Data 2005) was consulted. After getting four recent Satellite images, the animated loop can then be accomplished by setting the visibility. A function was then created to set visibility of two images. Following is the visibility function for animating the satellite images:

```
// set visibility of two images, image1 on, image2 off
function setVisibility(divOn, divOff) {
    div1 = document.getElementById(divOn);
    div2 = document.getElementById(divOff);
    div1.style.visibility='visible';
    div2.style.visibility='hidden';
}
```

The function above turns the visibility style of two objects into on and off accordingly by acquiring references to the object and then changing the visibility style of the object. To enable animated loop for satellite imagery, the main concept is to sets all satellite images visibility off except the latest one, then hide current visible image and unhide another image at certain interval. The visibility manipulation is done in `animate()` function. The `setTimeout()` function is used to invoke a function after certain interval. The animation concept is as following:

```
(Let image4 = latest, image3 = 1 hr before, image2 = 2hr
before and so on)
Initially, all images is hidden except the latest one
While ( stop = false ) {
      Wait one interval, set image 1 visible, image 4 hidden
      Wait one interval, set image 2 visible, image 3 hidden
      Wait one interval, set image 3 visible, image 2 hidden
      Wait one interval, set image 4 visible, image 1 hidden
      Wait two intervals
End while
```

The implementation result is not as expected. It was discovered during testing that the web browser is not waiting for the `setTimeout()` to be finished before executing next line. This can be understood because if the web browser is going to wait, the whole loading process will be really slow. Thus the timing for next image was increased by multiple of waiting interval. The looping is done by recursive call using `setTimeout()` function. Following is the final result of animate function for satellite imagery.

```
//function to animate images
function animate() {
    if (stop) { return; }

    // get images' ID in order of latest time
    // i.e. div[0] = recent, div[1]= image at one interval behind recent
    div = imageOrder();

    // all images (with z index = 5) start with hidden visibility
    // set visibility time out in sequence, i.e. animate
    setTimeout("setVisibility(div[0], div[3])",1*speed);
    setTimeout("setVisibility(div[1], div[0])",2*speed);
    setTimeout("setVisibility(div[2], div[1])",3*speed);
    setTimeout("setVisibility(div[3], div[2])",4*speed);
    setTimeout("animate()",5*speed);
}
```

Another problem which rooted from same principle was found. It is not necessary that each line is executed in sequence even the acquiring images are done in sequence in the HTML coding, thus the images may not in time order. A sorting function was then created to solve this problem. Following is the coding:

```
// function to sort images by returning image ID in order of time lag
function imageOrder() {
    // container for images sources and indices
    var imSrc = new Array(4);
    var indices = new Array(4);

    // get images sources
    imSrc[0] = document.images.S1image.src;
    imSrc[1] = document.images.S2image.src;
    imSrc[2] = document.images.S3image.src;
    imSrc[3] = document.images.S4image.src;
    var imArray = new Array(imSrc[0], imSrc[1], imSrc[2], imSrc[3]);

    // sort images by time, the image source are labelled by time
    imArray = imArray.sort();
    // get image order
    for (i = 1; i <= 4; i++) {
        for (j = 1; j <= 4; j++) {
            // get image ID of latest
            if (imSrc[j-1] == imArray[i-1]) {
                indices[i-1] = 'S' + j;
            }
        }
    }
    // return image ID
    return (indices);
}
```

First, an array is created. The image source location (pathname) is then stored inside this array. A copy of the array is then created and the content is sorted by using `sort()` method of array. For the new sorted array, a lookup or search is then performed to determine the sequence (index) of sorted array. These sequence are then stored in a new array and returned by the function.

Since there is a problem with image order, the initial visibility setting for the image must be changed too. Following is the code for it:

```
// set image order for layering of images
function sortDiv() {
    //if all images are downloaded, set the image order
    if(document.images.completed) {
    div = imageOrder();
    div1 = document.getElementById(div[3]);
    div1.style.visibility='visible';
    div4 = document.getElementById(div[0]);
    div4.style.visibility='hidden';
    div2 = document.getElementById(div[1]);
    div2.style.visibility='hidden';
    div3 = document.getElementById(div[2]);
    div3.style.visibility='hidden';
    } else {
    // if images are not finished download, wait for 0.2 second
    setTimeout('sortDiv()',200);
    }
}
```

This function obtains sorted image ID at beginning, then the image visibility for all satellite images is set accordingly. The concept is that if the images are not finished downloaded, wait four 0.2 second and then examine the images again. If all images are downloaded, set their visibility so that the latest satellite image is the visible one.

After solving the problem, the remaining task is created simple functions as follow:

- `startLoop()` – set ‘stop flag’ equals to false;
- `stopLoop()` – set ‘stop flag’ equals to true;
- `speedUp()` – decrease interval time; and
- `speedDown()` – increase interval time.

These functions are invoked when the button in the web design is clicked by user. A refresh button is also included to update the series of satellite images. Overall, the functions are working properly and no error messages were found by the JavaScript Console. The satellite images obtained are confirmed to be latest available ones.

3.2.2 Lightning Strike Information

During the research, the best Lightning strike information that can be accessed was Energex's Lightning Tracker. The storm map displays cloud-to-ground lightning strikes in southeast Queensland during the past 45 minutes. ENERGEX does not own the lightning information provided by GPATS. They are bound by a contract with GPATS allowing us to display limited lightning information. Under the agreement, Energex must delay the publication of Lightning Tracker data to the Internet by 15 minutes (Energex 2005). In project development, more concern was then addressed on copyright issue.

Furthermore, for each update, the lightning strike map is stored as different filename. The naming for these updates (files) is not of certain format or timestamp. Further research is carried out to solve the naming process but in vain. In addition to the copyright issue, it was later decided that it is better not to include Energex's Lightning Tracker in the web interface design. The alternative is to provide a link to Energex's Lightning Tracker website.

3.2.3 Rain Radar Map

For Mount Kent Observatory, Marburg Rain Radar Map provided by Bureau of Meteorology was the best rain radar map found in the research. It can be downloaded freely from the website. The direct address for the image was found later by scanning through the page source. The address is:

http://mirror.bom.gov.au/radar/IDR503.gif.

The radar map is sampled every 10 minutes and received approximately 4 minutes after the sampling (Bureau of Meteorology Australia 2005a). In brief, the delay is about 4 to 14 minutes. Since the image is updated regularly and the pathname is still the same, there is no need to work out the pathname for latest image.

Furthermore, copy of the images is stored with timestamp detailing up to minutes and this figure is not following a certain pattern. Thus it is difficult to obtain the latest series of images except if we go and track for each minute. In the website design, the animated loop for Rain radar map was excluded.

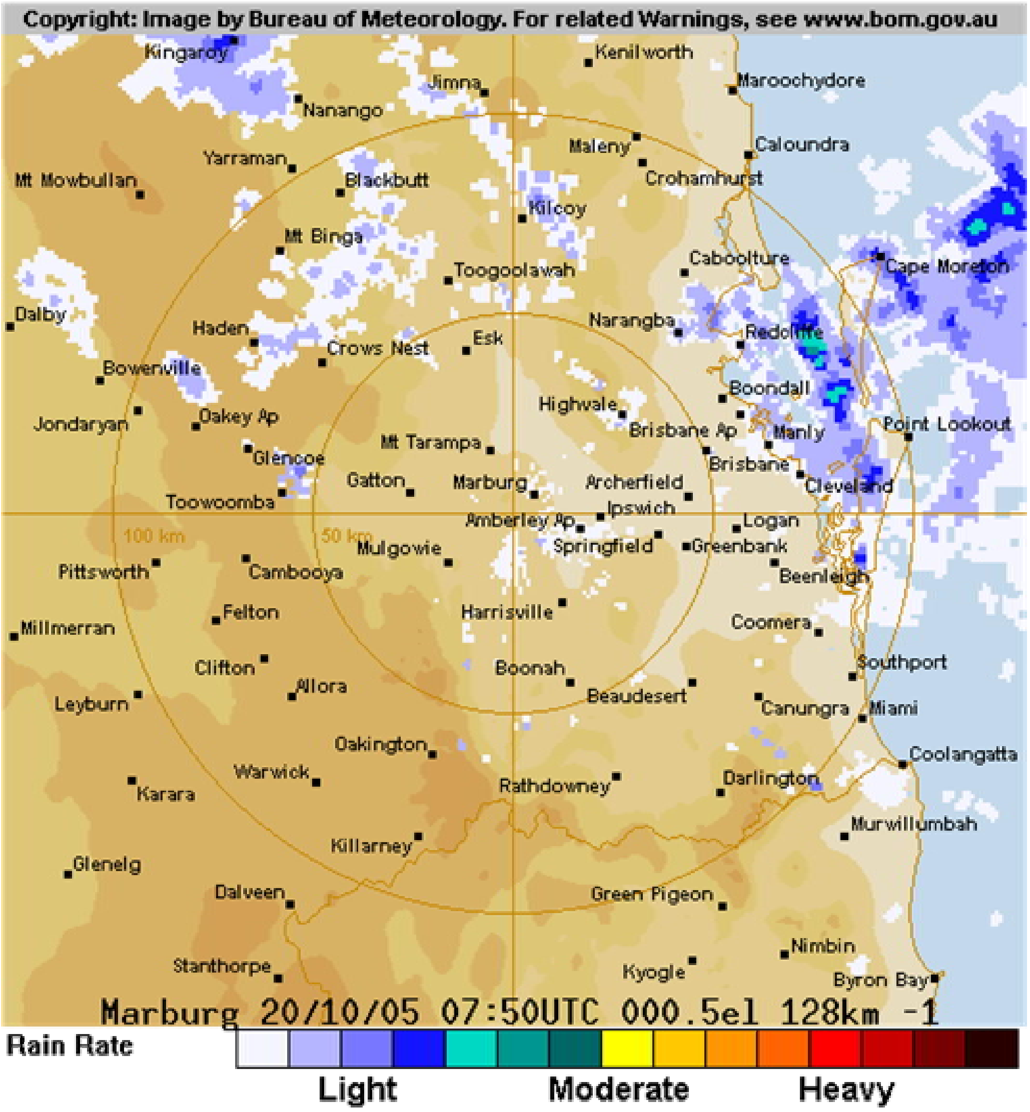

Figure 3.2: Brisbane (Marburg) Rain Radar (adapted from Bureau of Meteorology Australia (2005a))

3.3 Mt Kent Pinpoint Design

After successfully obtained and implemented the weather maps, the next step was to find Mt Kent location within the weather map and pinpoint it. There were only two weather maps to be considered as lightning tracker is not included. For Marburg rain radar, more details Southern Queensland map is required. To find Mount Kent location in Rain radar map, the nearest cities are chosen as references. A search on most detail and appropriate map for Mt Kent Observatory was carried out. Following is the map acquired from online map by Geoscience Australia (2004)

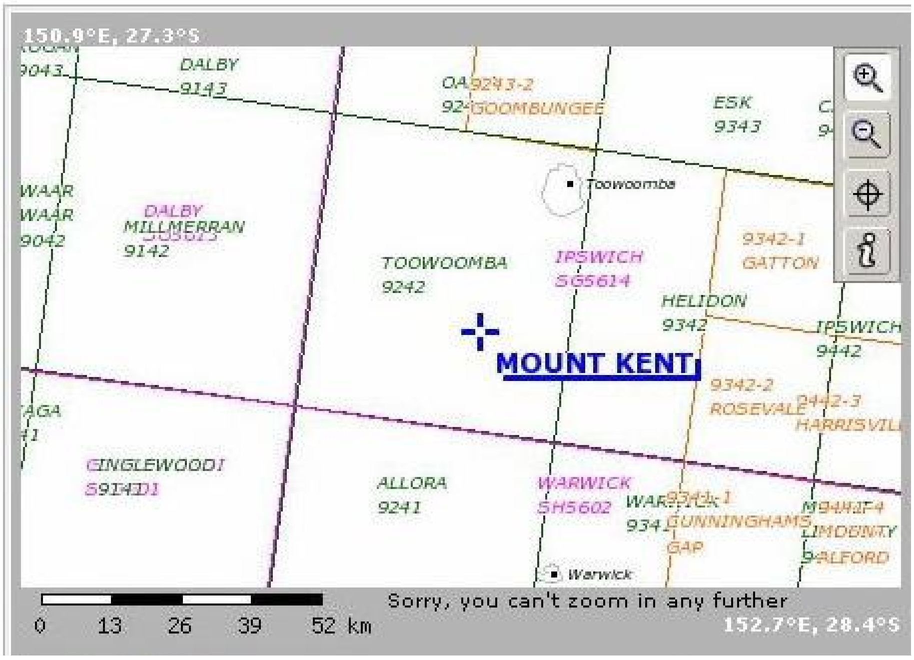

Figure 3.3: Mount Kent Map (adapted from Geoscience Australia (2004))

The map above is then compared with Marburg rain radar map as in figure 3.2. It can be comprehended that the cities which appropriate to be chosen as reference points are Toowoomba and Warwick. A simple mathematic calculation was then applied to find the location of Mount Kent in rain radar map. Following is the concept:

1. Print both maps;
2. Draw a line in Mt Kent map to connect Toowoomba (T) and Warwick (W);
3. From the line, draw a perpendicular line to Mt Kent (K);
4. Find the distance from Mt Kent to the perpendicular corner;
5. Repeat step 2 and 3 for Marburg rain radar map;
6. Find the location of Mt Kent in the Marburg rain radar map by using ratio;
7. Find the X(top) and Y(left) for Mt Kent in printed rain radar map in ratio to the entire map; and
8. Using the ratio of X and Y to the map, calculate the exact X and Y for Mt Kent in the image file itself (in pixels).

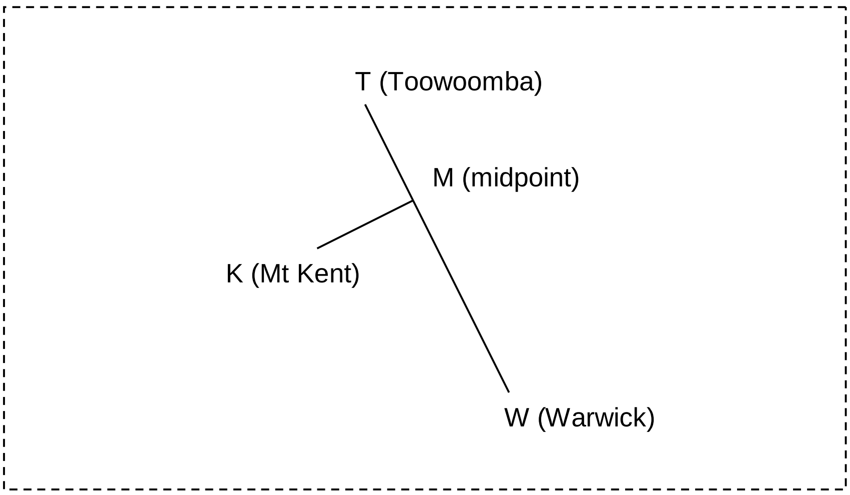

Figure 3.4: Finding Mount Kent location. Line KM is perpendicular to line TW

The calculation is shown in following table:

Table 3-1: Estimating Mount Kent Distance

Mt Kent Map (print)		Marburg Rain Radar Map (print)	
Route	Distance	Route	Distance
T to W	5.2 cm	T to W	3.85 cm
T to M	2.0 cm	T to M	(3.85/5.2) x 2.0 = 1.48 cm
W to M	3.2 cm	W to M	(3.85/5.2) x 3.2 = 2.73 cm
K to M	1.1 cm	K to M	(3.85/5.2) x 1.1 = 0.82 cm

Table 3-2: Estimating Mount Kent Location

Mt Kent location (print)		Mt Kent location (IDR503.jpg)	
Coordinate	**Distance(ratio)**	**Coordinate**	**Distance (pixel)**
X (top)	3.00/13.52	X (top)	(3.00/13.52) x 524 = 116
Y (left)	8.00/14.55	Y (left)	(8.00/14.55) x 564 = 310

After finding the top and left of Mount Kent, an image at same resolution with Marburg Rain Radar was created with white background and a pinpoint drawn on exact estimated Mt Kent location (pixel). *ACD Photo Editor 3.1* was used to create the pinpoint. This image was than converted to transparent background by using giftrans.exe, a Debian package for DOS. Following command was run in DOS command prompt to create a copy of the image with transparent background:

```
giftrans -t 1 pinpoint.gif > pinpoint2.gif
```

The command above creates `pinpoint2.gif`, the same image with `pinpoint.gif` except that the background is transparent. The command manual are obtained form an online manual reference page (Ley 1994). This transparent pinpoint image is then overlayed on top of radar map.

For coloured infrared satellite image, the accuracy of locating Mt Kent is less since the map is in bigger scale. The only way to pinpoint it is to compare the satellite image with the rain radar map, and approximating the Mt Kent location by referring to noticeable boundary: Queensland border and east coast lines. The pinpoint for Satellite image was also created with same resolution as satellite image and then overlayed on top of satellite images.

3.4 Image Positioning

Overlay is used in the web interface design so that the browser has less workload when the weather maps are refreshed. The refresh function will update the weather maps and the pinpoint remained pinned on top of the maps. In studying overlaying image, a number of internet searches, programming and testing had been done. The overlay was done by using CSS positioning style and JavaScript to determine the location of overlay. DIV element (an HTML object) is used to contain the image and a class is created to implement absolute positioning style for the DIV element. Following is the DIV class for overlay image (in CSS):

```
DIV.initial{ position:absolute; top:0px; z-index:2; visibility:hidden; }
```

The z-index presents the layers. Higher layer is overlay on top of lower one. The images was set to hidden initially to hide them before positioning.

Following JavaScript functions are used to find position of an object in HTML document (i.e. use to find weather maps location).

```
// find left(X) position of an object
function findPosX(obj){
    var curleft = 0;
    if (obj.offsetParent){
        while (obj.offsetParent){
            curleft += obj.offsetLeft
            obj = obj.offsetParent;
          }
    }else if (obj.x)
          curleft += obj.x;
    return curleft;
}
```

The code above is used to find X position of an object. Firstly, it set the variable `curleft` to 0, If the browser supports `offsetParent`, go into a loop that continues as long as the object has an `offsetParent`. Inside the loop, add the `offsetLeft` of the element relative to the offsetParent to `curleft` and set the object to this `offsetParent`. The while loop repeats this process as long as the element has an `offsetParent`. When it has no more offsetParent, the HTML element is reached and we have the position relative to it. As to browser supporting `x`, take the `x` property of the link (Koch 2005). Same code is used for finding Y position except that `offsetLeft,` and `x` is replaced by `offsetTop,` and `y` respectively.

A function is then created to set an object position to be the same with another one. The coding is as follow:

```
function setOverlay(sourceID, targetID) {
    source = document.getElementById(sourceID);
    target = document.getElementById(targetID);
    var newX = findPosX(source);
    var newY = findPosY(source);
    target.style.top = newY + 'px';
    target.style.left = newX + 'px';
}
```

This function gets a hold of the `source` and `target` object. Then the X and Y for the `source` object are acquired and assigned to `target left` and `top`. Thus `target` will be removed from its original location to the same location as the `source`.

Since all overlay images are cointained in DIV.initial objects which visibility was set to hidden at initially, there is a need to reset the visibility for all satellite images. The function `sortdiv()` as discussed before in chapter 3.4 is reinvoked at the beginning to sort the images and set their visibility accordingly.

Overall, the image positioning coding is running as expected after intensive testing and debugging process is carried out. It was verified to work properly with both Internet Explorer and Mozilla.

3.5 Refresh function

Since the pinpoint is stored as different image to the weather maps, the refresh (update) process for radar map and satellite imagery can be easily done by updating the image sources for the weather maps. However, even it seemed that it is an easy job, it did not work as expected in early progress. The first attempt was to reset the image source to the same pathname but it did not work. A lot of time had then been spent to find a way to refresh a single image or HTML object but in vain. Then it was decided to use a page refresh function. The page refresh JavaScript code by Grizzly Webmaster (2001) is quite robust. Below is the code:

```
<noscript>
<!--
    We have the "refresh" meta-tag in case the user's browser does
    not correctly support JavaScript or has JavaScript disabled.

    Notice that this is nested within a "noscript" block.
-->
<meta http-equiv="refresh" content="2">

</noscript>

<script language="JavaScript">
<!--

var sURL = unescape(window.location.pathname);

function doLoad()
{
    // the timeout value should be the same as in the "refresh" meta-tag
    setTimeout( "refresh()", 2*1000 );
}

function refresh()
{
    //  This version of the refresh function will cause a new
    //  entry in the visitor's history.  It is provided for
    //  those browsers that only support JavaScript 1.0.
    //
    window.location.href = sURL;
}
//-->
</script>

<script language="JavaScript1.1">
<!--
function refresh()
{
    //  This version does NOT cause an entry in the browser's
    //  page view history.  Most browsers will always retrieve
    //  the document from the web-server whether it is already
    //  in the browsers page-cache or not.
    //
    window.location.replace( sURL );
}
//-->
</script>

<script language="JavaScript1.2">
<!--
function refresh()
{
    //  This version of the refresh function will be invoked
    //  for browsers that support JavaScript version 1.2
    //  The argument to the location.reload function determines
    //  if the browser should retrieve the document from the
    //  web-server.  In our example all we need to do is cause
    //  the JavaScript block in the document body to be
    //  re-evaluated.  If we needed to pull the document from
    //  the web-server again (such as where the document contents
    //  change dynamically) we would pass the argument as 'true'.
    //
    window.location.reload( false );
}
//-->
```

```
</script>
</head>

<!--
    Use the "onload" event to start the refresh process.
-->
<body onload="doLoad()">
```

The refresh function above is a good practice as it includes functions for different version of JavaScript and also for those browsers that do not support JavaScript. This page refresh function updates the whole page including the weather maps. However, it relocates the current display focus to the top of the webpage. The relocation will certainly disappoint user. Thus, the research went back to concentrate on updating single element or image. The solution was discovered later during research on setting up live webcam. An image can be updated by resetting the source as following example:

```
image1.src='http://studentweb.usq.edu.au/Webcam.jpg?' +
             Date.parse(new Date().toString())
```

The refresh single image problem was then solved. For Satellite imagery, there is no need to use this method as the algorithm for getting recent image from FTP server has been implemented.

3.6 Design Layout

Generally, a web interface design shall present well even for different display resolution. Different user may have different screen resolution. The common display resolution used nowadays are 1024 x 768 pixels while some still used 800 x 600 pixels. Higher resolution will be preferred in future since the price of LCD monitor is now cheaper and affordable. The website was designed accommodate this issue. In addition to this, the use of colour, font, and other details in the web design shall be appropriate too.

The background used for the website is black as it gives less eyestrain. The black colour may also represent the night sky for astronomer. The font used in the website design is courier new as it is simple and easy to read. The font size is not of fixed size, it was set to small, medium, large, extra large, etc so that the font size is determine by the web browser at client computer. The webpage has simple layout with header or title, navigation column, content and footer When the display windows is resize, the layout is shall also resized to fix the window if possible (it is not wise to squeeze the image or font, but the paragraph width can be reduced and so with some HTML element). The styling for the website was mostly done in Cascading Style Sheet (CSS).

During the design, it was found that in some web pages, the content is not enough to fit whole display window, which results in leaving the footer hanged in the middle of the screen. To solve this problem, two approaches are used: JavaScript function and CSS manipulation. The JavaScript function was used for webpage with navigation list while CSS is for the alternative.

For those with navigation list, the overall height is compose of, header, spacer (which separate header and body), navigation column, and footer height. A JavaScript function can then be used to set this height to fit display window. This function will then be invoked when the document is loaded and resized. Below is part of the code:

```
function setElementPos(navID, footerID) {
    element1 = document.getElementById(navID);
    element2 = document.getElementById(footerID);
    var totalHeight =document.body.offsetHeight;
    var totalWidth =document.body.offsetWidth;
    totalWidth = totalWidth - 10 + 'px';
    totalHeight = totalHeight - 190 + 'px';
    element1.style.height = totalHeight;
    element2.style.width = totalWidth;
}
```

The function above executed as following sequence:

1. Get reference to elements that contains navigation list and footer
2. Get current display window height and width
3. Calculate the suitable height for navigation list and width for footer
4. Set the height and width for elements that contain navigation list and footer respectively

The code works properly with Internet Explorer and Mozilla, but when it was tested on student web page provided by USQ, the code cause Internet Explorer to stop responding (hang). Mozilla still functioning well when the setElementPos() is invoked. There was no solution found to solve this problem.

For webpage without navigation list, CSS setting is used instead of implementing a similar JavaScript function for the content (in substitute to navigation list). This is to reduce the coding of the webpage. The concept is to wrap the whole HTML document with DIV elements and set their height to fit display window. Following is the illustration for using DIV elements:

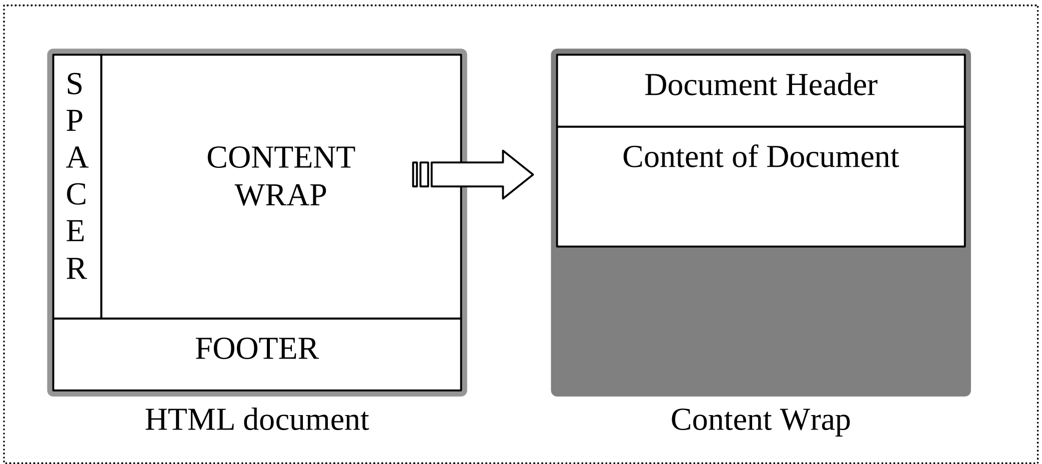

Figure 3.5: Footer positioning concept using DIV element and styling format.

`SPACER`, `CONTER WRAP` and `FOOTER` are DIV elements, while the website header and content is store as table encapsulated within `CONTENT WRAP`. The height for `SPACER` is set to 94%, leaving 6% of height for `FOOTER`, the width of `CONTENT WRAP` is set to 99% leaving 1% for `FOOTER`. This height and width setting was then implemented in CSS and the result was not as expected. It was discovered later that the spacer must has at least some value to content with. After its margin was set to one pixel, it spans 94% of overall height. To conclude, this concept and its implementation are working properly.

Apart from footer problem, we need to concern about navigation between web pages. The main idea is to satisfy user by providing access to link page without closing or exiting main page. This is commonly used in presenting image gallery, and linking to external website. To validate this, a pop up (open new) window is used. Following is the HTML code for opening disclaimer page in new window:

```
<a href="Disclaimer.html" onClick="window.open('Disclaimer.html',
'myWin', 'status, resizable=no, width=350, height=300'); return false">
<U>Disclaimer</U></a> statement.
```

When the link is clicked, the window.open() is invoked to open new non-resizable window for "Disclaimer.html" with size of 350 x 300 pixels and status bar, `'myWin'` is the ID for new window. If the same ID is used for the thumbnails link, there will be only one popup window used apart from the main page.

Overall, web interface design is quite challenging. A syntax error is not easy to be detected within the code without advance debugging tool. Furthermore, the design must also supportable by both Mozilla and Internet explorer. During the design development, a lots of problems appeared especially due to web browsers compatibility. All of the problems are solved except for the auto positioning for footer which is implemented in JavaScript.

Chapter 4

Remote Observing

Basically, when the idea of web based observatory is introduced, people will expect a webpage that provides a star or sky map from observatory viewpoint; some tools or buttons to move the telescope, operate the camera and focuser, and capture image and video; and a camera display that shows current scope movement.

This may seem amazing to start with. However, this is not the best way to control a remote observatory. Besides of wasting telescope time, it requires a continuous internet connection and login to assure that images or videos are acquired. The continuous connection exposes observatory equipment to more internet threat and also may have more program faulty during design development. Furthermore, it exposes observatory instruments to abuse or mistakes, and thus requires training and trust.

A secure and easy way to provide remote access of observatory to others is to provide fill-out forms that let users specify what they want and when, not how. The robotic observatory needs to be hands-off. This protects observatory instruments and computer, eliminates interruptions due to connection lossage, and eliminates the need to train users on how to aim, focus, and expose (Denny 2005b).

ACP provides the automation hub for the observatory. It sequences the observations and controls the telescope, CCD imager, filter wheel, auto-guider, focuser, and dome (Denny 2005a). ACP has optional built-in web browser and FTP access package: *Share Your Sky!*. It has been designed to operate *completely hands-off*. To acquire single image, users just need to fill out a form and submit it. An online deep-sky catalogue lookup is provided to assist form filling. However, for multiple images, *Share Your Sky!* uses a more efficient and robust architecture:

1. The observer first prepares a plan, a simple list of targets and parameters for acquiring images, then uploads it to the web based observatory where it is stored permanently in a private folder with other plans previously prepared by the observer.

2. Next the observer starts the processing of the plan. At this point, the entire process is under the control of the observatory computer. The observer can then log off the internet, confident that the plan will be executed. The next morning they can log back in and pick up their images.

The observer may watch the progress of the plan via a web page that displays a detailed *observing log* produced by the automation at the observatory. The observer can simply log off and return at any time later to check up on the progress of the run or retrieve images and the observing log when the run has been completed (Denny 2005b).

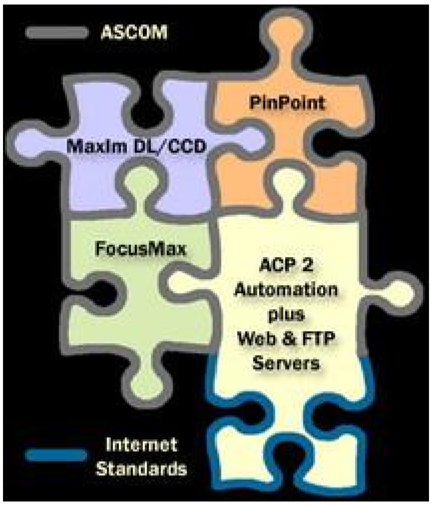

Figure 4.1 ACP automation plus web and FTP servers. ACP provide interface between internet standard and ASCOM platform

To examine ACP built-in web server and FTP access, the trial version of ACP, ASCOM (platform for ACP) and MAXIM DL were installed. The step by step web server and FTP access configuration is provided in “ACP Help”. Following part of this chapter will discuss the configuration and result.

The ACP was started and it took a few second. Below is ACP window on first run.

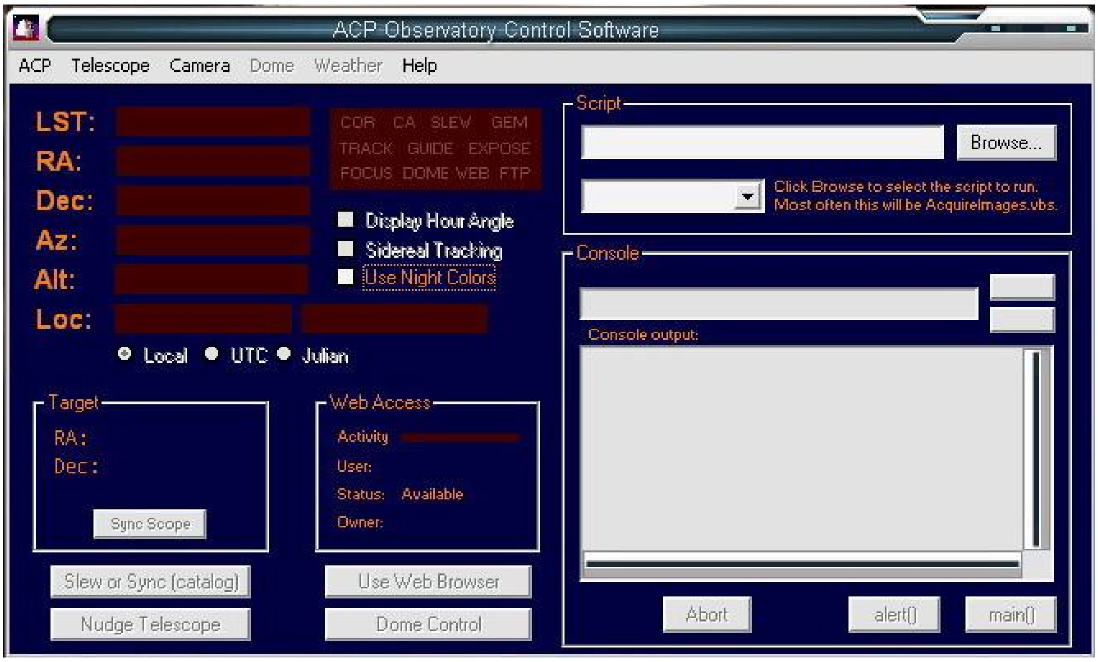

Figure 4.2: ACP window on first run. Most of the features are not available yet.

Most of the features are not functioning as the observatory instruments are not installed, configured or connected to the computer. The web server and FTP access can be enabled at ACP preferences server setting as follow:

1. Click *ACP* menu and select *Preferences*. This will show *preferences* window.

2. Click the *servers* tab (see Figure 4.3). Tick the enable web server and FTP server checkboxes, the port number can be input here. The default port number for HTTP and FTP are 80 and 21 respectively. The owner can set timeout for each user login session at here. The IP address shown here was dynamic IP address for the web and FTP server. The path for document root and log files is required to be input. Document root is the folder for ACP web server usage, i.e. all web page file, resources and server tools.

3. Click ok and the servers are enabled

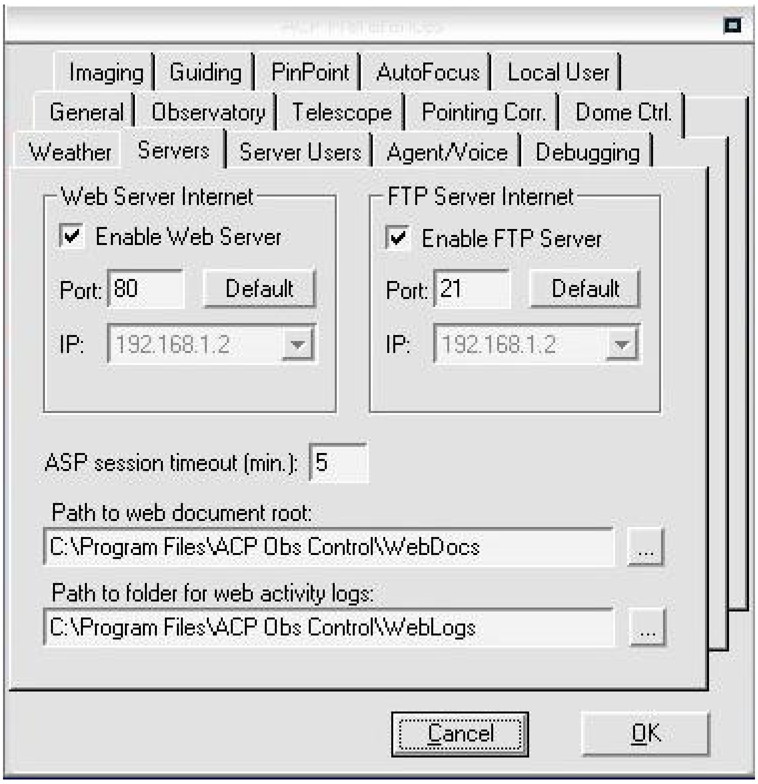

Figure 4.3: ACP Preference window

To run a server, a static IP address is required. Static IP was provided by Internet Service Providers (ISP). However, due to the need of an unlimited number of IP address, many ISP limit the number of static IP addresses they allocate and economize on the remaining by temporarily assigning an IP address (dynamic) to a requesting Dynamic Host Configuration Protocol (DHCP) computer from a pool of IP addresses. Most dialup Internet connection use dynamic IP address and the contrary for DSL connection.

After enabling the web server and FTP access, the next step is to register a user. To create a user, open the preferences window and select *server users* tab, click *new* button and enter user *full name* and *username*, and click *ok* (see Figure 4.4). A new user will be created and ACP will generate a random password which are difficult to remember and extremely difficult to crack. The operator can then click the options to allow certain feature such as script execution (for operating instruments) and FTP server.

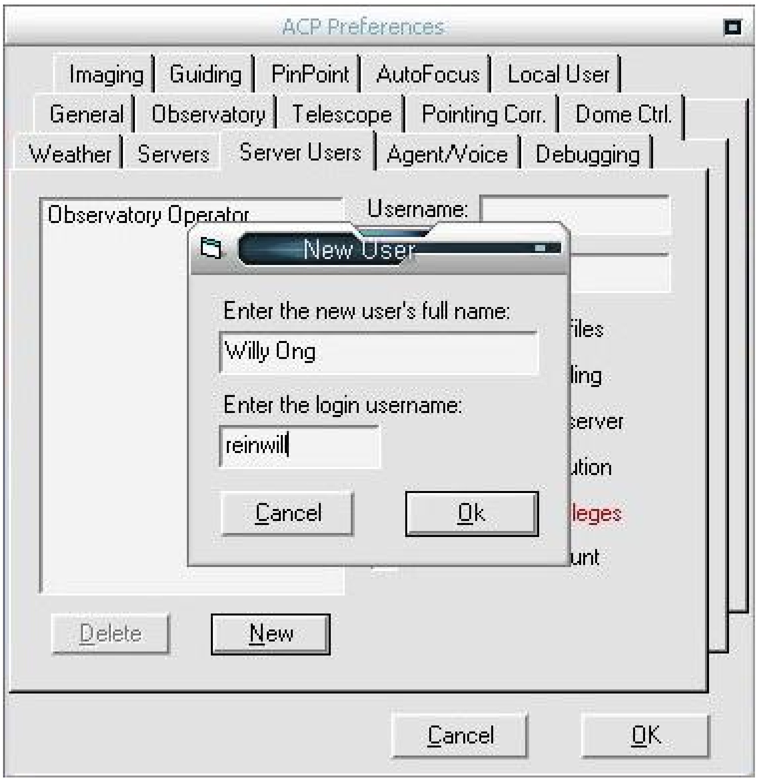

Figure 4.4: ACP preferences window – create user

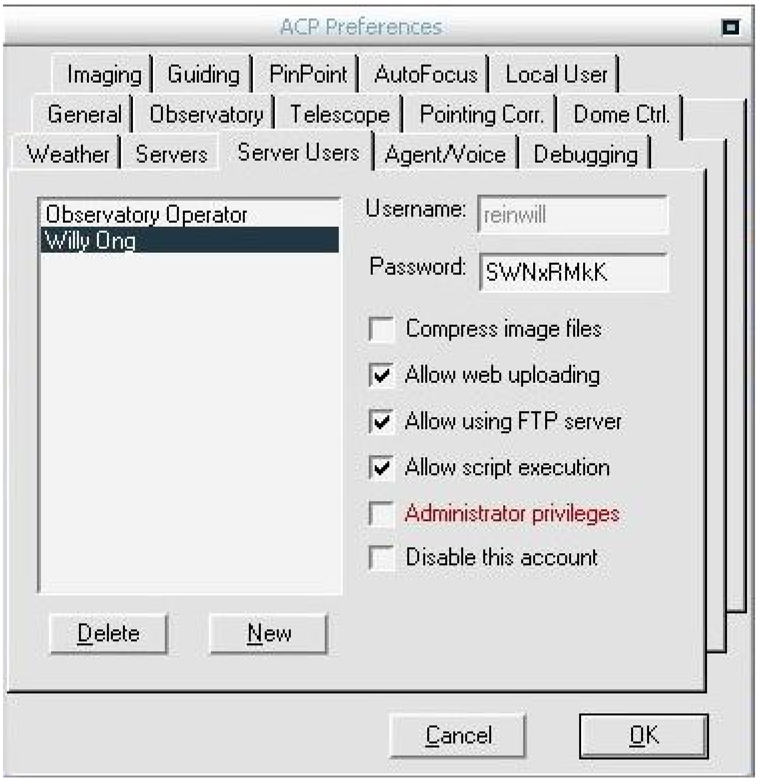

Figure 4.5: ACP preference window – User Setting

In the test run, the computer was connected to internet through a router modem: Aztech DSL600EU. A static address is given to the modem, and port forwarding is needed to be configured correctly so that the port will be forwarded to the specified host IP (i.e. the PC running web server). The DSL600EU doesn't support loop back address, which mean another external connection (not within the LAN) must be used to connect to the DSL600EU public IP address in order to test the port forwarding. The test has been done using external connection and the port forwarding is working properly. For simplicity, the remaining test is done on local computer, or dynamic IP address. Following is the testing procedures together with some screenshots.

Access web server - when accessing the web server, a popup window appeared for user login as figure below:

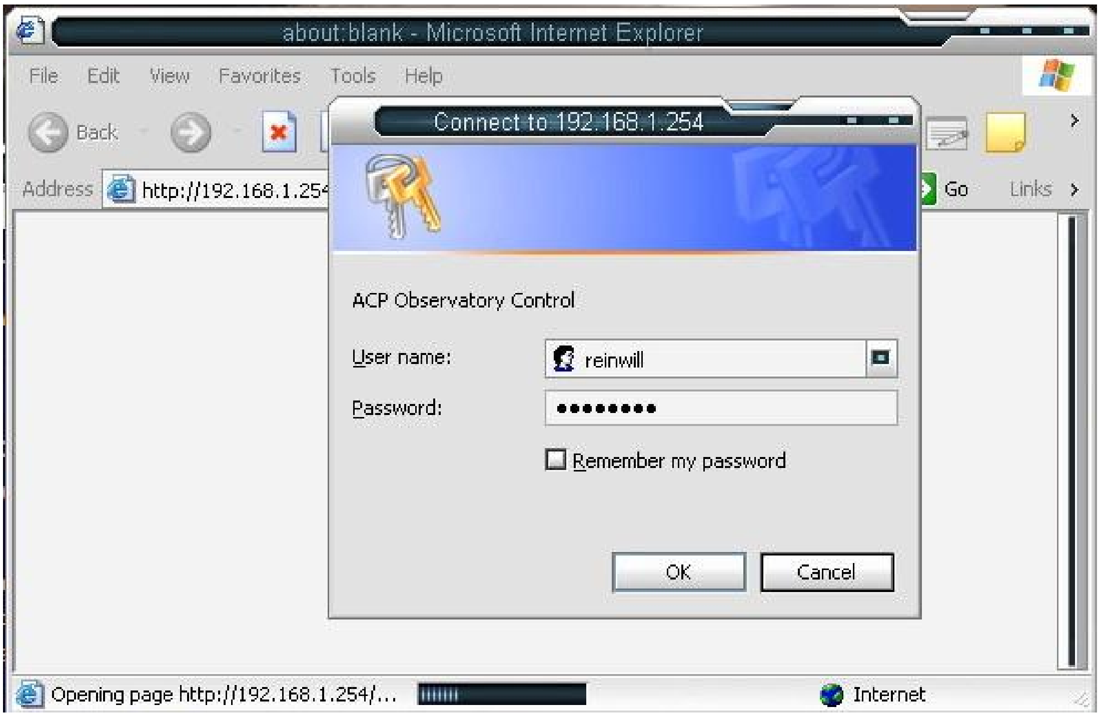

Figure 4.6: Accessing ACP web server

After the login succeeds, user was provided access to ACP web server. The web server provides many features, such as acquiring image, file management system, and help guide (see Figure 4.7). Acquiring image was not possible at this stage. The other tests that were carried out are: running scripts, adjusting image acquisition setting, upload files, and accessing common user file.

Some of scripts were working properly since they do not require any instrument installed. The output was shown in console panel of ACP program. Users can adjust their image acquisition parameters in *Adjust Your Setting*. A few files were copied into folder of ACP document root (i.e. local computer path for ACP web server) and during the test these files can be accessed through web browser.

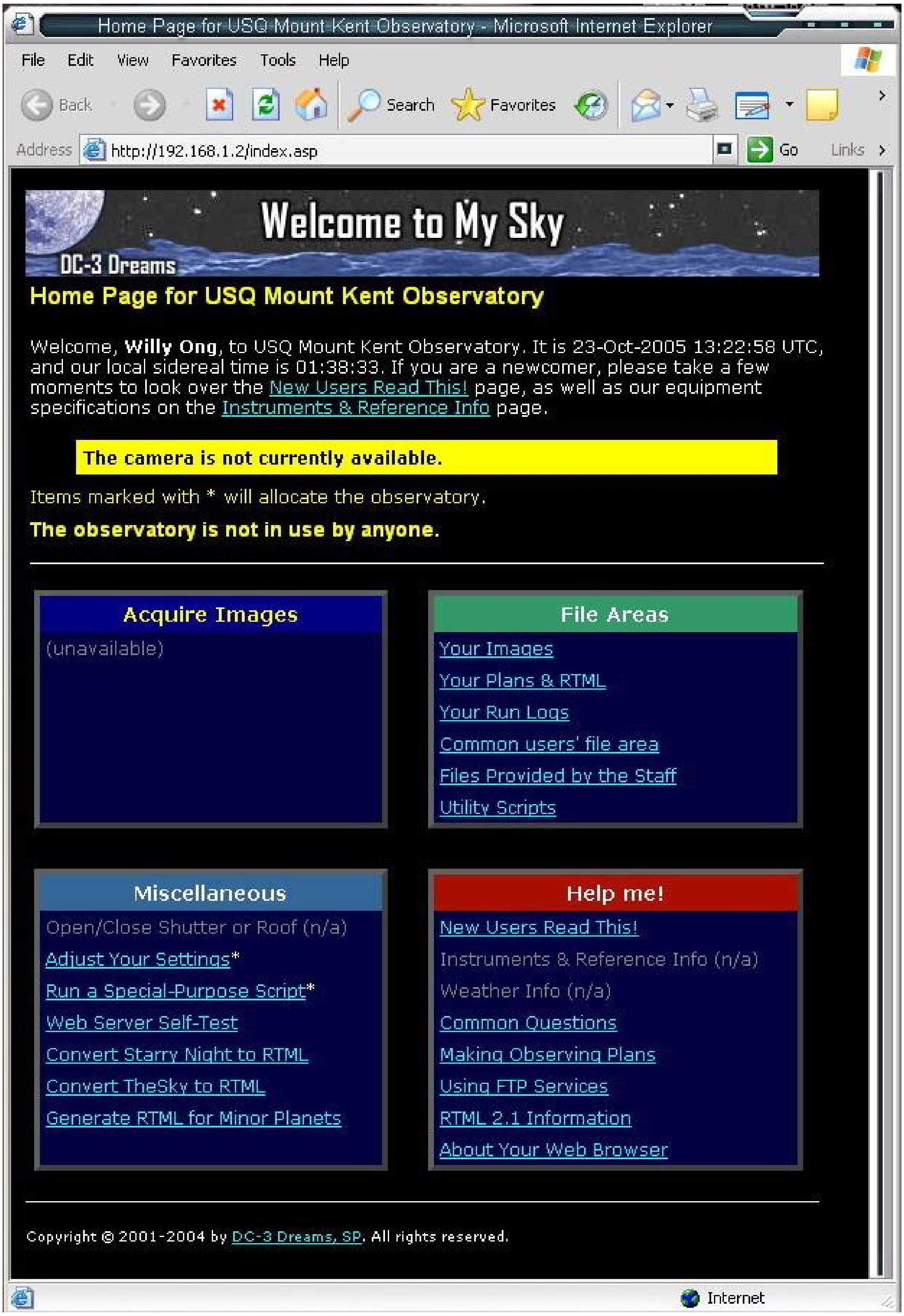

Figure 4.7: ACP website for remote imaging. Acquiring Images is unavailable since no telescope instruments were connected to the PC.

When the remote imaging is ready, user should be able to acquire image by setting plan as describe before in *Share Your Sky!* package. A more sophisticated planning can be done through ACP Planner which is currently free to download and use. ACP Planner™ (Patent Pending) gives user a visual way to create an observing plan within *Starry Night* or *TheSky*. This way of planning is a quantum leap beyond other as it allows user visualizes target positions and timing together dynamically and does task-oriented planning.

When user roll time back and forth (using the mouse wheel or Starry Night's Graph tool), targets move across the sky and a "guide display" shows whether there's an active observation at the time, or how long it will be until the next observation will start, or that the remainder of the night is free. Thus, user can easily visualize which targets are in a favourable position at what times, how much time is available to insert a target between ones already selected, or start a new one at the end of the list. When user finished adding targets and setting up observations, he/she can run the plan if available, or save it for later running or uploading to a remote ACP observatory (Denny, 2005b).

In conclusion, ACP provides remote observing through its built-in web server. The web server had been tested and it works appropriately except the remote imaging process (not available). The security is satisfactory as the operation of observatory instruments is done only through processing plan. In future, when USQ remote observing with ACP planner is validated, this will be a remarkable achievement.

Chapter 5

Webcam Study

Planetary photography using webcam is an interesting topic nowadays together with live display for telescopic image. This chapter will discuss webcam research, especially on live webcam. The live webcam was implemented inside the web interface design (*telescope.html*). In future application, other video input device can be used instead of webcam. The webcam used in the project was provided by Dr. Brad Carter and the model is Logitech Quickcam Express. Theoretically, this webcam is not suitable for planetary photography because it does not have CCD chip installed in it. Moreover, no telescope had been used in the research due to the circumstances. Nevertheless, this research was conducted with main purpose of setting live display which can be utilised for future live observing. Thus, hardware resource is not much to be concerned compared to the implementation.

The software supplied with the webcam will normally allow an AVI video file to be saved to hard disc. Logitech QuickCam Express comes with software featuring support for MSN, Yahoo and Windows Messenger live video call, video recording, and image capturing. The video and image capturing is the necessary function for planetary photography. For better result in planetary photography using webcam, processing software is used to process the captured video and image.

5.1 Planetary Photography using Webcam

The popular softwares for processing Webcam image are: *AstroSnap, RegiStax*, and *IRIS*. In following research, *RegiStax* was used to process planetary image and video. The raw image and video used in this section were downloaded from websites. Following is the raw image of moon captured on 9th August 1998 by Peck (1998) using Logitech QuickCam VC,

Figure 5.1: Raw Moon image (adapted from Peck (1998))

The image above was later processed by *RegiStax*. Using wavelet transform and layering feature, following is the result.

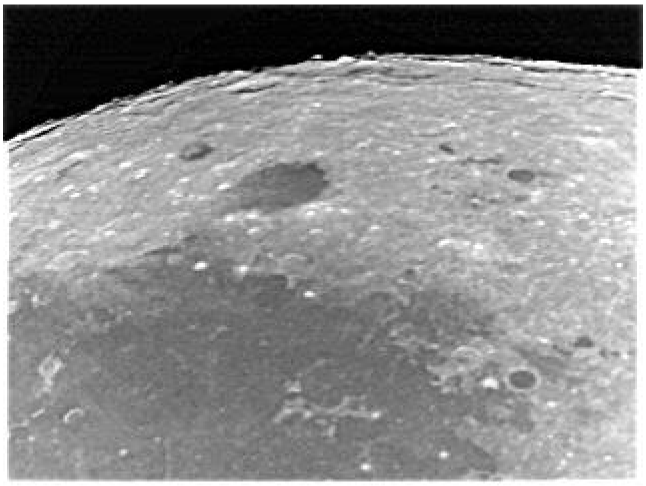

Figure 5.2: Processed Moon image using *RegiStax*

The processed image (Figure 5.2) seems much clearer and better than the raw image (Figure 5.1). *RegiStax* works well in image processing. The next stage is to use the stacking feature of *RegiStax*. To accommodate this, an internet search was performed on raw planetary video file. The video file will be used for creating better quality image by stacking frames together. Following is the first frame of a video file of Jupiter downloaded from Stellar Product website:

Figure 5.3: First frame of Jupiter video file adapted from (Stellar Products (2004))

This Jupiter video was taken by Donald G. Bruns (Stellar Products 2004) on 9th April 2004 with 10" Newtonian telescope, stopped down to 9.5"; 2.5x and 2x Barlow lenses to get F/33; Philips ToUcam used at 10 frames per second, 100 msec exposures, on slightly hazy skies. Only the first 50 frames of the video are provided in the website. This video file was processed further with *RegiStax*. *RegiStax* has auto aligning feature (using fast fourier transform) which help a lot in aligning the frames together for stacking. The stacking was also done automatically by *RegiStax*. The final two processing steps that provided by *RegiStax* are wavelet transform filter and image resizing, cropping and HSL (Hue, Saturation, and Lightness) adjustment. Following is the resulting image:

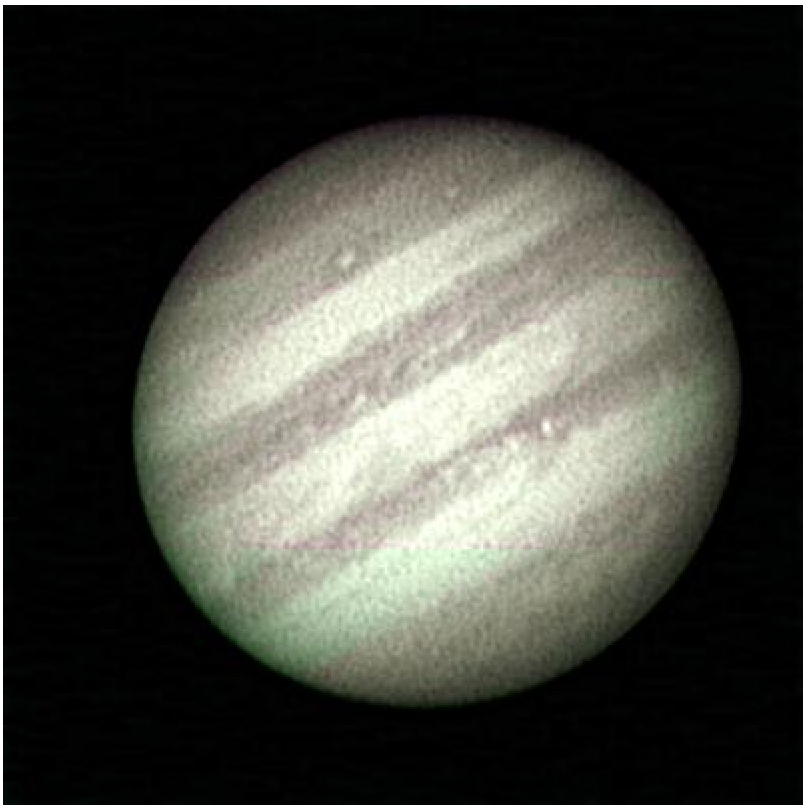

Figure 5.4: Jupiter Image produced from stacking 50 frames

The image above is quite a surprising result for planetary photography. This explains why planetary photography using webcam and processing software is popular nowadays.

The Jupiter image (Figure 5.5) processed by stacking of 669 frames of same video file using KCCD Tools is included after this for comparison purpose. This image was obtained from *Stellar Products* (2004). It is really a remarkable planetary photography produced from webcam video and processing tool. In conclusion, webcam is an inexpensive but promising imaging hardware for planetary photography.

Figure 5.5: Jupiter image produced by stacking 669 frames, adapted from (Stellar Products (2004))

5.2 Live Webcam

The main idea to implement live webcam in the web interface design is to validate internet-viewable display for live telescopic image. The basic concept used for live webcam implementation is as follow:

1. Have a program that interfaces with video input (webcam), captured the video input and stored it into an image file, and update the image file at regular interval.
2. The image file location should be inside document root of web server for the website.
3. Implement an auto refresh program in the web interface design to update image display regularly.
4. Start the web server for web interface design and run the webcam imaging program.

Designing a program to interface with webcam, capture, store, and update image regularly is a big task that would not be accomplished within the time constraint. The alternative is to look for software, preferably freeware, which perform all required task. In the research, two programs were selected carefully to perform this task: *TimerShot* (powertoys for windows XP) and *CoffeeCup Webcam3.5*.

TimerShot is a free *powertoys* package for windows XP platform. It is capable of capturing image from webcam or other video input device regularly and saved it into an image file. Timershot can update the captured image at shortest interval of one second. The image storing location can be set to the network places such as local network and FTP. However, during testing process, it was found that *TimerShot* has a few bugs. The program will stop updating captured image after a certain interval. In testing with update interval of two second, *TimerShot* stopped updating approximately 3 minutes after it started. Furthermore, the FTP uploading was not working properly. It fails to send the updated image to FTP server.

CoffeeCup Webcam 3.5 on the other hand provides more stable functionality than Timer shot. This software cost a total of $30.00 for purchase. A 30-day trial version was used in the project development. CoffeeCup provides more sophisticated features as TimerShot. It can be used to capture image and video. It has image setting feature of adding caption, changing exposure, file saving details and compression. The advance FTP setting helps much in testing the live webcam. During testing procedure, CoffeeCup Webcam3.5 performs well in updating images and FTP uploading.

CoffeCup Webcem 3.5 excels in every aspect when compare to *TimerShot*. It also has scheduler for capturing webcam image. In summary, CoffeeCup Webcam 3.5 is the feasible choice to perform the updating live webcam image. Following is screenshots of Timer Shot and CoffeeCup Webcam3.5.

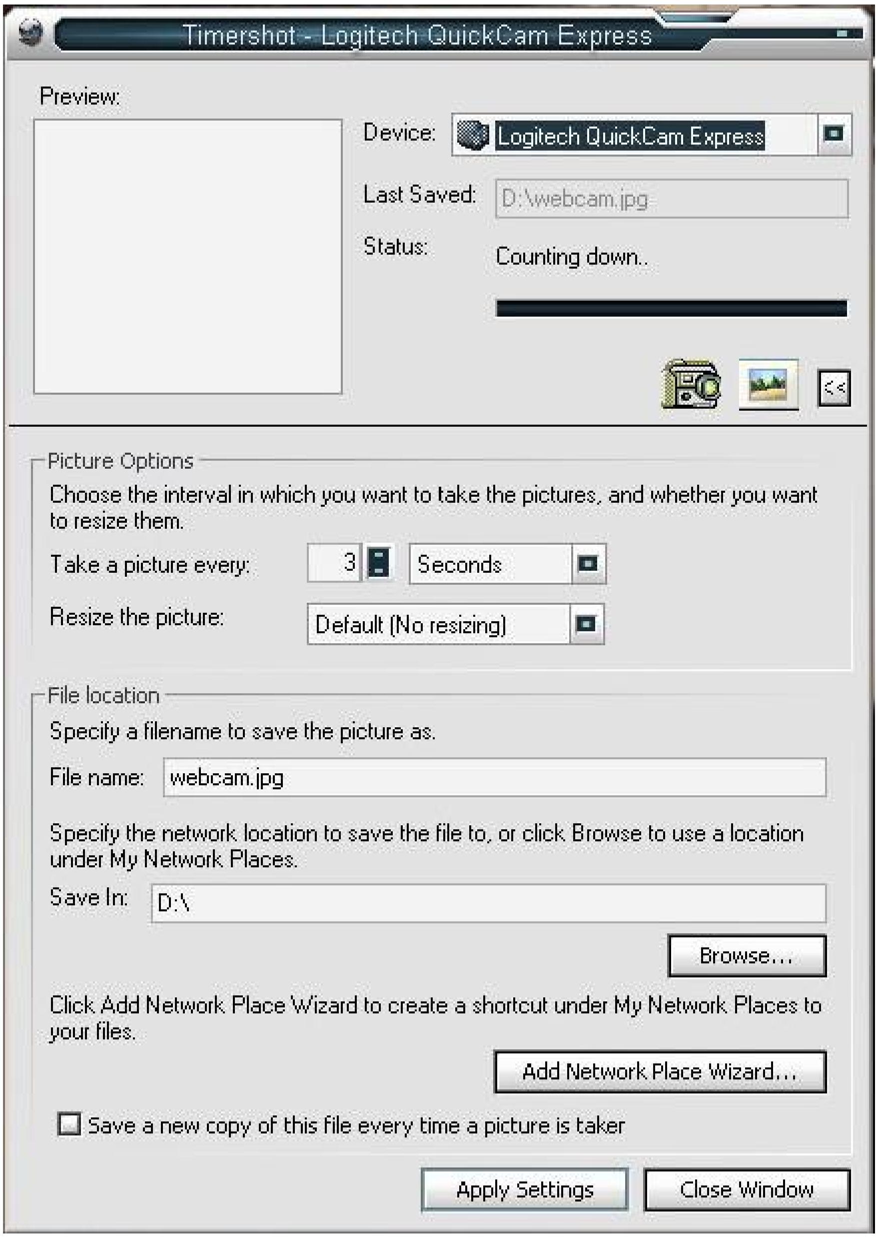

Figure 5.6: Timershot – Powertoys for Windows XP, used with Logitech QuickCam Express to captured and update *webcam.jpg* every 3 seconds

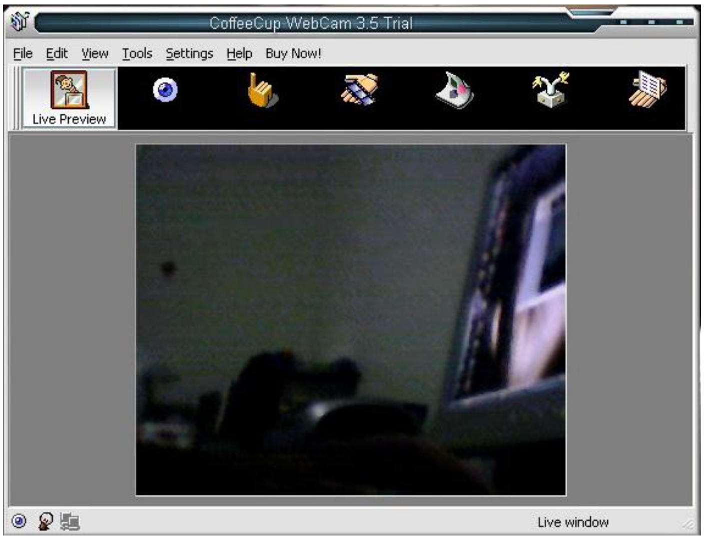

Figure 5.7: CoffeeCup Webcam 3.5. Logitech Quickcam Express was connected and the video captured is shown in the live preview window

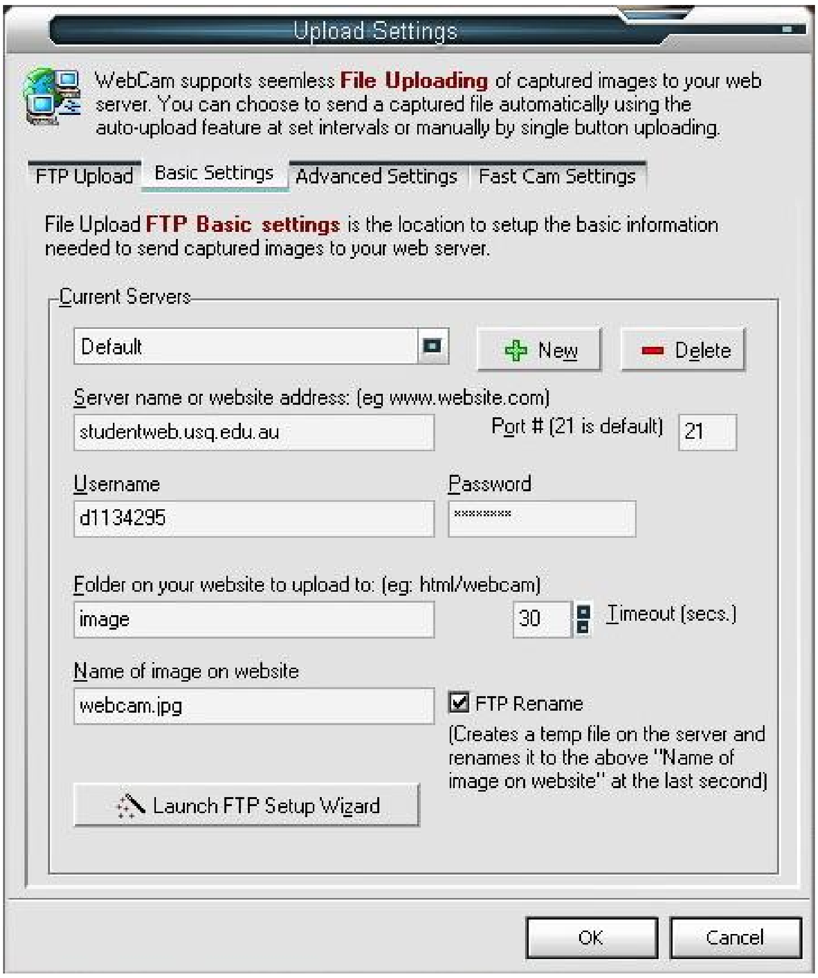

Figure 5.8: CoffeCup Webcam 3.5 FTP Upload Setting.

The implementation for live webcam in the web interface (HTML coding) was done by using timer function setTimeout() and clearTimeour() which was discussed before in chapter 3.4. The implementation in JavaScript is as follow:

```
function UpdateClock() {
   // clear timeout of previous setTimeout
   if(clockID) {
      clearTimeout(clockID);
      clockID  = 0;
   }
   //refresh webcam image
   refresh();
   // set timeout for recursive call of next iteration of update function
   clockID = setTimeout('UpdateClock()', 2000);  //refresh webcam image every 2 sec
}

function KillClock() {
   // clear timer
   if(clockID) {
      clearTimeout(clockID);
      clockID  = 0;
   }
}

function refresh() {
    //set image source to latest modified image
    document.images.webcam.src='image/webcam.jpg?' + Date.parse(new Date().toString())
}
```

The function `UpdateClock()` invokes the `refresh()` function. The refresh function implemented is similar to the Rain Radar Map one. `UpdateClock()` will call itself recursively after two second. `clockID` is the identification for each timer set. To avoid multiple timer, `UpdateClock()` will clear any previous timer before proceed to refreshing image and recursive call to itself by using `clearTimeout()`. `UpdateClock()` is called when the webpage for live webcam (telescope.html) is loaded, and `KillClock()` is invoked when exiting.

After finish the implementation, several tests were carried out together with *TimerShot* and *CoffeeCup Webcam 3.5*. The test on live webcam was conducted both on local computer, local network, and also on USQ student webpage. The entire web interface design was uploaded to student webpage (ftp://studentweb.usq.edu.au/home/d1134295) to test external access to the live webcam and also to test FTP uploading feature of *TimerShot* and *CoffeeCup Webcam 3.5*.

The program works well in displaying live webcam when tested on local computer and local network. For external access test, *TimerShot*'s FTP uploading was not working and it also stopped itself after certain interval.

In summary, the webcam study was enjoyable. The result of webcam research was promising. Image stacking is a good approach in planetary photography. The live webcam can be utilised further for live observing with staff running the observatory.

Chapter 6

Conclusion

The development of this project has been a long journey, but finally with the time limit, this project is needed to be drawn to a conclusion. In brief, the web interface design is working properly. It works with Internet explorer 6 and Mozilla 1.7.12. It was tested mostly on student webpage to verify external access. The working version is available on: http://studentweb.usq.ed.au/home/d1134295. The screenshot for the web interface design are included as Appendix D.

Moreover, there are two important topics that needed to be considered. First is the web server. Since ACP already has its own web server, it is not practically possible to set up another web server on same computer. This implies that for the web interface, another web server (computer) will be required. However, this is also reasonable as we are not going to use the live webcam with O'Mara telescope. The website can continue remained in student webpage while the computer that connected to the webcam can run an FTP uploading program to enable the live preview.

Another important topic is the copyright issue. The weather maps are obtained from Bureau of Meteorology Australia. Bureau of Meteorology permits any used of their materials for purposes of study, research, criticism and review. If the materials are reproduced, re-used or redistributed for any commercial purposes or distributed to the third party, a permission from Commonwealth of Australia is required (Bureau of Meteorology Australia (2005b)). Displaying the weather maps in the web interface also implies distribution or publication. Thus, a permission request was applied for the weather maps. The request was granted and the permission letter is enclosed as Appendix C. After dealing with the copyright issue, it become obvious that we need to include a copyright and disclaimer notice is necessary to be included in the web interface design. This is the reason why a footer containing copyright noticed is designed for every web pages in this project, and a disclaimer page is included as popup window.

The copyright issue wraps up the research project. In researching and developing this project, many existing skills have been recalled, as well as many new skills being developed from scratch such as creating website, live webcam and running a web server. The field trip to Mt Kent was enjoyable and more interest in astronomy was developed.

6.1 Achievement of objectives

Most of the objectives that were specified for this project have been successfully achieved. Unfortunately, there are several tasks that were unable to be implemented by the dead-line. Following is the details on objectives or activities to be completed according to project specification and the achievement

Research information related to software which controls Mt Kent Observatory equipment, including the telescope, camera, and the dome itself.
This topic has been accessed and presented in chapter 2.1. A further research on ACP is presented in chapter 4.

Research the communications infrastructure available to Mt Kent Observatory and USQ, including the Apache web server for the observatory

This topic has been accessed and presented in chapter 2.1. No further research on it is required as the Observatory already has internet access.

Investigate server-side PHP scripting for the web server

During project development, there is no server side application needed for the web interface design. HTML coding, CSS and JavaScript are enough for the web interface design. This task is therefore neglected.

Research the CCD image capture software (camera-PC interface)

The camera and PC interface is managed by Maxim DL. ACP interface with Maxim DL for remote observing and image capturing. Since the control is carried out by ACP, there is no further research required. Alternatively, in webcam research, the interface between the webcam and PC has been assesed. This is presented in Chapter 5.1.

Design an Internet–viewable display of recent telescopic images captured by an SBIG camera

Since ACP licence was purchased for use with O'Mara Telescope instruments (include the SBIG camera) and ACP has its own web server, this objective cannot be carried out with the SBIG camera. On the other hand, a study on webcam for planetary photography has been done. An internet-viewable live webcam display was implemented on the web interface design. The live webcam implementation is discussed in Chapter 5.2.

Design a web interface which encapsulates: recent satellite imagery, lightning strike information, and rain radar for Mt Kent, CCD camera (current) image and camera image gallery

This objective has been accessed and implemented. The detail for the web interface designed is presented in chapter 3.

Include weather station output and cloud sensors in the web interface design (optional)

As discussed before, the equipment is not ready yet. Therefore, this activity was not carried out during project development.

Research remote access to software that controls the telescope and security aspects (optional)

The remote access is entirely under ACP control. ACP web server provides secure user access and account managed. This was addressed in Chapter 4.

Undertake a security audit of the Mt. Kent technology infrastructure including firewalls, operating system and software (optional)

This task was not accessed in detail. The reasons are: firstly, time limit; secondly, the internet threats are mostly related to remote access and ACP already have password login security; ACP remote image is using plan submission rather than continuous login, this eliminates most of the threats; finally, the observatory's internet connection are through university network (include firewall protection), therefore, it is more secure.

6.2 Further Work

There is only a few works left for this project since the remote observing is handled by ACP web server. Once the O'Mara Telescope is ready for remote observing, ACP web server can be started, and a link to ACP web page can be added to `telescope.html`. For live webcam, it can be use directly with any available instrument. Further work is needed to solve the auto positioning problem. The main task left is to include weather station and cloud sensor output in the web interface design. The output shall be the most recent one and updated frequently. This activity required more research and knowledge on interface between weather station, cloud sensor and PC.

Glossary of Terms

ACP Astronomy Control Panel

BOM Bureau of Meteorology (Australia)

CCD (Charge Couple Device) A light-sensing device containing grids of pixels which is used in digital cameras, optical scanners and video cameras for recording images. They are more efficient than photographic film, in capturing incident light.

CSS (Cascading Style Sheet) A stylesheet language used to describe the presentation of a document written in a markup language.

DIV An element or structure in HTML document which is used to sets the default horizontal alignment for the block-level elements it contains.

IMG An element if HTML document use for containing and displaying image

FreeBSD A free, open source, Unix-like operating system descended from AT&T UNIX via the Berkeley Software Distribution (BSD) branch. FreeBSD is developed together as an entire operating system.

FTP (File Transfer Protocol) Software standard for transferring computer files between machines with widely different operating systems. It belongs to the application layer of the Internet protocol suite.

HTML (HyperText Markup Language) A Standard Markup language designed for the creation of web pages and other information viewable in a browser.

HTTP (HyperTest Transfer Protocol) the client-server TCP/IP protocol used on the World-Wide Web for the exchange of HTML documents.

IP Address A number used by computers to refer to each other on the Internet. Each networking hardware that connects to internet is allocated specific IP address, either static of dynamic.

MKO Mount Kent Observatory

OS (Operating System) System software that responsible for the direct control and management of hardware and basic system operations. Additionally, it provides a foundation upon which to run application software such as word processing programs and web browsers, eg: Windows 2000, Windows XP, Linux RedHat 7.

Pathname The fully specified name of a file in a computer, including the position of the file in the file system's directory structure.

Port number Number that used to identify port (channel) for different internet application programs. E.g. FTP and HTTP has default port (21 and 80 respectively) assigned to them

Protocol A set of formal rules describing how to transmit data, especially across a network. E.g. Transmission Control Protocol and Internet Protocol

Wavelets refer to the representation of a signal in terms of a finite length or fast decaying oscillating waveform (known as the mother wavelet). This waveform is scaled and translated to match the input signal.

List of References

Bureau of Meteorology, Australia 2005a, *128km Brisbane (Marburg) Radar*, Commonwealth of Australia, Brisbane, viewed 8 April 2005, <http://mirror.bom. gov.au/products/IDR503.shtml>

Bureau of Meteorology, Australia 2005b, *Copyright Notice*, Commonwealth of Australia, Brisbane, viewed 28 April 2005, <http://www.bom.gov.au/other/ copyright.shtml>

Bureau of Meteorology, Australia 2005c, *Satellite Images*, Commonwealth of Australia, Brisbane, viewed 28 April 2005, <http://www.bom.gov.au/weather /qld/>

Carter, B 2005, *The Solar-Stellar Connection*, powerpoint presentation, University of Southern Queensland, Toowoomba. Viewed 21 September 2005, <http://www. usq.edu.au/users/carterb/DSP/index.htm>

Denny, R B 2005a, *ACP Observatory Control Software*, DC3-Dreams, Mesa, viewed 19 May 2005, < http://acp4.dc3.com/index2.html>

Denny, R B 2005b, *Share Your Sky! – Your Observatory on the web*, DC3-Dreams, Mesa, viewed 19 May 2005, < http://acp4.dc3.com/index2.html>

Department of Physics 2004, *Digital Science Partnership - A Proposal to the National Aeronautics and Space Administration in Response to an Education Program Earmark*, University of Louisville, Louisville.

Diffraction Limited 2004, *Maxim DL: The Gold Standard*, Diffraction Limited, Ottawa, viewed 18 May 2005, <http://www.cyanogen.com/products/maxim_main.htm>

Geoscience Australia 2004, *Place Names Search: MOUNT KENT*, Commonwealth of Australia, Canberra, viewed 2 May 2005, <http://www.ga.gov.au/bin/gazd01?rec=139574>

Grizzly Webmaster, 2001, *JavaScript: Page Refresh/Reload*, Bear Consulting Group, viewed 30 August 2005, <http://grizzlyweb.com/webmaster/javascripts/refresh.asp#meta-tag>

Hillocks, J 2004, *Mt Kent Observatory*, Faculty of Science, University of Southern Queensland, viewed 15 May 2005, <http://www.usq.edu.au/sciences/research/casrac/ astronomy/mtkent.htm>

Koch, P P 2005, *JavaScript - Find position*, Amsterdam, viewed 4 August 2005, <http://www.quirksmode.org/js/findpos.html>

Kurkowski, T 2005, 'Using Webcam for Planetary Photography', *Schmidt Cassegrain telescope advice - planetary astrophotography with a webcam*, viewed 16 May 2005 <http://www.sctscopes.net/Photo_Basics/Webcams_for_Planets /webcams_for_planets.html>

Ley, A 1994, 'Manual Reference Pages - GIFTRANS', *FreeBSD Man Pages*, GSP Services, Inc, Bowie, viewed 5 September 2006, <http://www.gsp.com/cgi-bin/man.cgi?section=1&topic=giftrans>

Peck, A1998, *Moon Imaging with a QuickCam*, San Jose, viewed 1 September 2005, <http://www.shallowsky.com/quickcam/oldpix.html>

Refsnes Data 2005, *JavaScript References* 2005, viewed 25 July 2005, <http://www.w3schools.com/jsref/default.asp>

Software Bisque, Inc. 2005a, *Automadome: Dome Control Software*, Software Bisque, Inc., Colorado, viewed 18 May 2005, <http://www.bisque.com/Products/AutomaDome/>

Software Bisque, Inc. 2005b, *The Sky6 for Windows*, commercial website, Software Bisque, Inc., Golden, viewed 18 May 2005, <http://www.bisque.com/v6/TheSky_Version_6.htm>

Stellar Products 2004, *Jupiter 2004*, Stellar Products, San Diego, viewed 10 September 2005, <http://www.stellarproducts.com/imageall/jupiter/jup2004/jup2004.htm>

The Open Source 2004, *PHP Comparison to JavaScript*, Imagenation, Ontario, viewed 8 August 2005, <http://www.theopensourcery.com/phpcomparetojs.htm>

Wikimedia Foundation, Inc. 2005, *WIKIPEDIA: The Free Encyclopedia*, San Diego, viewed 9 August 2005. <http://en.wikipedia.org/wiki/Main_Page>

Appendix A

Project Specification

University of Southern Queensland
Faculty of Engineering and Surveying

ENG4111/4112 Research Project
PROJECT SPECIFICATION

FOR:	**Willy ONG**
TOPIC:	Remote Monitoring and Instrumentation for Mt Kent Observatory
SUPERVISOR:	Dr. John Leis Dr. Brad Carter, Faculty of Science
SPONSORSHIP:	Faculty of Engineering and Surveying, USQ Faculty of Science, USQ
ENROLMENT:	ENG4111 – S1, D, 2005 ENG4112 – S1, D, 2005
PROJECT AIM:	The project aims to create a web interface design that provides remote observation and instrumentation for the USQ-maintained observatory at Mt Kent, by presenting recent or current telescope camera images, weather information for Mt Kent, weather station output, and if possible to provide remote control of the telescope

PROGRAMME: Issue A, 6th April 2005

1. Research information related to software which controls Mt Kent Observatory equipment, including the telescope, camera, and the dome itself.
2. Research the communications infrastructure available to Mt Kent Observatory and USQ, including the Apache web server for the observatory
3. Investigate server-side PHP scripting for the web server.
4. Research the CCD image capture software (camera-PC interface).
5. Design an Internet–viewable display of recent telescopic images captured by an SBIG camera.
6. Design a web interface which encapsulates:
   – Recent satellite imagery, lightning strike information, and rain radar for Mt Kent
   – CCD camera (current) image and camera image gallery

As time permits:

7. Include weather station output and cloud sensors in the web interface design.
8. Research remote access to software that controls the telescope and security aspects.
9. Undertake a security audit of the Mt. Kent technology infrastructure including firewalls, operating system and software

AGREED:

_________________ (Student) _________________ , _________________(Supervisors)

Date: _____ / _____ / _______

Appendix B

Code Listing

B.1 `index.html` HTML Code

`index.html` is the HTML code for the main page of Mt Kent Observatory. It includes introduction to Mt Kent Observatory, UTC and local clock, weather maps: Marburg Rain Radar and Coloured infrared with Mt Kent pinpointed. And link to Energex Lightning Tracker.

Listing B.1 `index.html`

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
      "http://www.w3.org/TR/html4/loose.dtd">
<HTML>

<HEAD>

<script language="javascript">
 <!--
//=========================================================================================
//================================= Start of JavaScript ===============================

//****************************** Image Animation Functions ****************************
var stop = false;              // animation stop flag
var speed = 800;                    // animated loop speed

// set visibility of two images, image1 on, image2 off
function setVisibility(divOn, divOff) {
    div1 = document.getElementById(divOn);
    div2 = document.getElementById(divOff);
    div1.style.visibility='visible';
    div2.style.visibility='hidden';
}

//function to animate images
function animate() {
    if (stop) { return; }

    // get images' ID in order of latest time
    // i.e. div[0] = recent, div[1]= image at one interval behind recent
    div = imageOrder();

    // all images (with z index = 5) start with hidden visibility
    // set visibility time out in sequence, i.e. animate
    setTimeout("setVisibility(div[0], div[3])",1*speed);
    setTimeout("setVisibility(div[1], div[0])",2*speed);
    setTimeout("setVisibility(div[2], div[1])",3*speed);
    setTimeout("setVisibility(div[3], div[2])",4*speed);
    setTimeout("animate()",5*speed);
}

function startLoop() {
    stop=false;
    animate();
}

function stopLoop() {
    stop=true;
}

function speedUp() {
    speed = speed * 0.7;
```

```
}

function speedDown() {
    speed = speed * 1.3;
}

//******************************** Image Sorting Functions *****************************
// function to sort images by returning image ID in order of time lag
function imageOrder() {
    // container for images sources and indices
    var imSrc = new Array(4);
    var indices = new Array(4);

    // get images sources
    imSrc[0] = document.images.S1image.src;
    imSrc[1] = document.images.S2image.src;
    imSrc[2] = document.images.S3image.src;
    imSrc[3] = document.images.S4image.src;
    var imArray = new Array(imSrc[0], imSrc[1], imSrc[2], imSrc[3]);

    // sort images by time, the image source are labelled by time
    imArray = imArray.sort();
    // get image order
    for (i = 1; i <= 4; i++) {
        for (j = 1; j <= 4; j++) {
            // get image ID of latest
            if (imSrc[j-1] == imArray[i-1]) {
                indices[i-1] = 'S' + j;
            }
        }
    }
    // return image ID
    return (indices);
}

//***************************** Image Positioning Functions ****************************
// overlay or positioning target image on top of an source image
function setOverlay(sourceID, targetID) {
    source = document.getElementById(sourceID);
    target = document.getElementById(targetID);
    var newX = findPosX(source);
    var newY = findPosY(source);
    target.style.top = newY + 'px';
    target.style.left = newX + 'px';
}

// find left(X) position of an object
function findPosX(obj){
    var curleft = 0;
    if (obj.offsetParent){
        while (obj.offsetParent){
            curleft += obj.offsetLeft
            obj = obj.offsetParent;
         }
    }else if (obj.x)
         curleft += obj.x;
    return curleft;
}

// find top(X) position of an object
function findPosY(obj) {
    var curtop = 0;
    if(obj.offsetParent) {
         while(obj.offsetParent) {
             curtop += obj.offsetTop
             obj = obj.offsetParent;
         }
    }else if (obj.y)
        curtop += obj.y;
    return curtop;
}
```

```
// set image order for layering of images
function sortDiv() {
    //if all images are downloaded, set the image order
    if(document.images.completed) {
        div = imageOrder();
        div1 = document.getElementById(div[3]);
        div1.style.visibility='visible';
        div4 = document.getElementById(div[0]);
        div4.style.visibility='hidden';
        div2 = document.getElementById(div[1]);
        div2.style.visibility='hidden';
        div3 = document.getElementById(div[2]);
        div3.style.visibility='hidden';
    } else {
        // if images are not finished download, wait for 0.2 second
        setTimeout('sortDiv()',200);
    }
}

// set the width for an element to suit display window (for footer)
function setElementWidth(elementID) {
    element1 = document.getElementById(elementID);
    var totalWidth =document.body.offsetWidth;
    totalWidth = totalWidth - 10 + 'px';
    element1.style.width = totalWidth;
}

//******************************** Get Satellite Imagery ******************************
var count = 0;         // counter for image lag

// get satellite image of different lagging
function getLatestImage(target) {

    // get current datetime and create datetime with lag
    var now = new Date();
    milli = Date.parse(now.toString());
    lag = (1 + count)*3600000;              // extra one hour lag for publication delay
    milli -=lag;
    var nowLag = new Date(milli);

    // get specific string format for datetime
    var years= nowLag.getUTCFullYear();
    var months= nowLag.getUTCMonth() + 1;  // the month range from 0-11, thus add 1
    var dates= nowLag.getUTCDate();
    var hours= nowLag.getUTCHours();
    dd = mm = hh = "";
    if (months < 10) {                      // convert to two digit
        mm = '0';
    }
    if (dates < 10) {                       // convert to two digit
        dd = '0';
    }
    if (hours < 10) {                      // convert to two digit
        hh = '0';
    }

    // concat the datetime string format and create pathname
    timeValue = ''+ years + mm + months + dd + dates + hh + hours + '30';
    pathname = 'ftp://ftp2.bom.gov.au/anon/gen/gms/IDE00035.'+timeValue+'.jpg';
    // assign pathname to image source
    target.src = pathname;
    count++;
}


//*********************************** Initialisation **********************************
function initialSetting() {
    setElementWidth('footer');                       // set footer width
    clockID = setTimeout('UpdateClock()', 500);      // start clock
    RRPin = document.getElementById('RRP');          // set pinpoint visiblility on
    RRPin.style.visibility='visible';
    SatPin = document.getElementById('SP');
```

```
    SatPin.style.visibility='visible';
    setOverlay('RRI','RRP');                // set satellite images position
    setOverlay('S1','S2');
    setOverlay('S1','S3');
    setOverlay('S1','S4');
    setOverlay('S1','SP');
    sortDiv();                              // sort satellite images
}

//****************************** UTC and Local Time Display ****************************
var clockID = 0;
function UpdateClock() {
   // clear timeout of previous setTimeout
   if(clockID) {
      clearTimeout(clockID);
      clockID  = 0;
   }
   // get current datetime and assigned it to UTCTime and LocalTime object
   var tDate = new Date();
   document.theClock.UTCTime.value = tDate.toUTCString();
   document.theClock.localTime.value = tDate.toLocaleString();
   clockID = setTimeout("UpdateClock()", 1000);
}

function KillClock() {
   // clear timer
   if(clockID) {
      clearTimeout(clockID);
      clockID  = 0;
   }
}

//*********************************** Refresh function *********************************
// Refresh rain radar map
function refreshR() {
    document.images.RRimage.src='http://mirror.bom.gov.au/radar/IDR503.gif?'
    + Date.parse(new Date().toString())
}

// Refresh satellite imagery
function refreshS() {
   count = 0;                                       // reset lag counter
   getLatestImage(document.images.S1image);         // get 4 latest image
   getLatestImage(document.images.S2image);
   getLatestImage(document.images.S3image);
   getLatestImage(document.images.S4image);
   setTimeout('sortDiv()',200);                     // sort images and set visibility
}

//================================== End of JavaScript =================================
//======================================================================================
//-->
</script>

  <!--set title, stylesheet, author, keyword for search, and description of document-->
  <TITLE>USQ Mt Kent Observatory - Home</TITLE>
  <LINK REL=stylesheet HREF="site.css" TITLE="Site Style Sheet" TYPE="text/css">
  <META NAME="author" CONTENT="Willy Ong">
  <META NAME="keywords"
        CONTENT="USQ astronomy observatory Toowoomba Mount Kent Australia">
  <META NAME="description"
        CONTENT="Main page for USQ observatory at Mt Kent">
</HEAD>


<!--************************************* HTML BODY **********************************-->
<!--set function call when loading, resizing and unloading document-->
<BODY onload="initialSetting()" onResize="initialSetting()" onunload="KillClock()">

<!--*********************** WEBPAGE HEADER **************************-->
<!--display header for Mt Kent Observatory Website-->
```

```
<TABLE ID="header" background="image/stripes.gif">
  <TR ID="header" >
    <TD ID="left" rowspan="2">
      <img ALT="USQ logo" src="image/usqlogo.gif"></TD>
    <TD ID="center"> <H1> Mount Kent Observatory </H1></TD>
    <TD ID="right" rowspan="2" background="image/trails_300px.jpg">
      <img src="image/stripes2.gif"></TD></TR>

  <!-- navigation for USQ home, usq search and contact us web pages-->
  <TR ID="header" >
    <TD id="link"> <a class="box" href="http://www.usq.edu.au">USQ Home</a> |
                   <a class="box" href="http://www.usq.edu.au/search">Search</a> |
                   <a class="box" href="contactus.html">Contact Us</a></TD></TR>
  <!--rule for seperating header and body-->
  <TR class="rule"><TD colspan="3"></TD></TR>
</TABLE>

<!--************************ WEBPAGE BODY ***************************-->
<TABLE>
  <TR class="space"></TR>
  <TR>
    <!--************* site navigation *************-->
    <TD id="nav" rowspan="10"><UL id="site-nav">
        <LI><a href="index.html">Home</a></LI>
        <LI><a href="telescope.html">Telescope</a></LI>
        <LI><a href="gallery.html">Gallery</a></LI>
        <LI><a href="temp.html">Weather<BR> Station</a></LI>
    </UL></TD>

    <!--************** Introduction ***************-->
    <TD id="padding" colspan=2>
        <H4>Welcome to USQ Mt Kent Observatory </H4><BR>
        Mt Kent Observatory is a facility for astronomy teaching, research and
        outreach.The Observatory is at a dark-sky site located about 30km southwest
        of Toowoomba and its three domes each house a telescope. In addition to the
        telescopes, the Mt Kent site houses the Educational Development Group (EDG)
        building with a lecture room, accommodation facilities and a control room for
        indoor operation of the observatory telescopes. A weather station is also
        located on the Observatory grounds. A high speed data link from the Observatory
        to the USQ campus has been installed for remote operation of the robotic
        telescope.<BR><BR><BR><BR>

        <H3>Mount Kent's Weather Condition</H3><BR>
        Following images are the latest Rain Radar and Satellite Imagery from Bureau
        of Meteorology, Australia. Marburg Rain Radar map is sampled every 10 minutes
        and received approximately 4 minutes after the sampling. In brief, the delay is
        about 4 to 14 minutes. The scanning of coloured infrared satellite image takes
        approximately 30 minutes to complete and the image is generally available
        within 30 minutes of completion. Thus, the minimum delay in satellite image
        is about one hour and the maximum is 2 hours. Energex's Lightning Tracker are
        delayed approximately 15 minutes. The timestamp for Energex Lightning Tracker
        is in local time. For radar map and Satellite image, the timestamp is in
        UTC/GMT format. The current UTC and local time are shown as follow:<BR><BR>
        <form name="theClock">
            UTC Time   = <input type=text name="UTCTime" size=40><BR>
            Local Time = <input type=text name="localTime" size=40>
        </form><BR></TD></TR>

  <!--********** Rain Radar Map Display ***********-->
  <!-- Title and Heading -->
  <TR><TD id="padding"><TABLE id="RainRadar">
    <TR><TD width=200px><H4>Rain Radar</H4></TD>
        <TD width=324px align="right">Mt. Kent <img src="image/RadarButton.gif"></TD>
    </TR>
    <!--Image Display-->
    <TR><TD ID="RRR" colspan=2 bgcolor=#aaaaaa>
            <div id="RRI">
            <img id="RRimage" src="http://mirror.bom.gov.au/radar/IDR503.gif"
                 alt="Marburg Rain Radar - Mt Kent"></div>
            <div id="RRP" class="initial">
            <img src="image/RadarPin.gif" alt="Marburg Rain Radar - Mt Kent"></div>
    </TD></TR>
```

```
    <!--footer: link and refresh button-->
    <TR><TD width=500px><a href="http://mirror.bom.gov.au/weather/radar/radarexp.shtml">
                           <U>Interpreting Weather Watch Radar Image</U></a></TD>
        <TD align="right"><BUTTON onclick="refreshR()" type="BUTTON">Refresh</BUTTON>
    </TD></TR>
  </TABLE></TD></TR>

  <TR class="space"></TR>
  <TR class="space"></TR>
  <TR class="space"></TR>

  <!--******** Satellite Imagery Display **********-->
  <!-- Title and Heading -->
  <TR><TD id="padding"> <TABLE id="Satellite">
    <TR><TD><H4>Satellite Imagery</H4></TD>
        <TD align="right">Mt. Kent <img src="image/RadarButton.gif"></TD></TR>
    <!--Image Display-->
    <!--All 4 latest Satellite Imagery are prepared-->
    <TR><TD colspan=2 bgcolor=#aaaaaa>
            <div id="S1"><img id="S1image" src="" onError="getLatestImage(this);"
                              alt="Satellite Imagery- Mt Kent"></div>
            <div id="S2" class="initial">
                 <img id="S2image" src="" onError="getLatestImage(this);"
                 alt="Satellite Imagery- Mt Kent"></div>
            <div id="S3" class="initial">
                 <img id="S3image" src="" onError="getLatestImage(this);"
                 alt="Satellite Imagery- Mt Kent"></div>
            <div id="S4" class="initial">
                 <img id="S4image" src="" onError="getLatestImage(this);"
                 alt="Satellite Imagery- Mt Kent"></div>
            <div id="SP" class="initial">
            <img src="image/SatellitePin.gif" alt="Satelite imagery - Mt Kent"></div>
    </TD></TR>
    <!-- Buttons for animate loop and refresh-->
    <TR><TD>
          <BUTTON onclick="startLoop()" type="BUTTON">Loop</BUTTON>
          <BUTTON onclick="stopLoop()" type="BUTTON">Stop</BUTTON>
              
          <BUTTON onclick="speedUp()" type="BUTTON">speed +</BUTTON>
          <BUTTON onclick="speedDown()" type="BUTTON">speed -</BUTTON></TD>
        <TD align="right">
          <BUTTON onclick="refreshS()" type="BUTTON">Refresh</BUTTON></TD>
    </TR>
    <!--Acknowledgement-->
    <TR><TD colspan=2>
       MTSAT-1R : Satellite image originally processed by the Bureau of Meteorology
       from the geostationary satellite MTSAT-1R operated by the Japan Meteorological
       Agency.
    </TD></TR>
  </TABLE></TD></TR>

  <TR class="space"></TR>
  <TR class="space"></TR>
  <TR class="space"></TR>


  <!--******* Link to Energex Rain Radar **********-->
  <TR><TD id="padding"> <TABLE id="Satellite">
    <TR><TD><H4>Lightning Tracker</H4>
     <BR>See Energex's Lightning Tracker at:
     <a href="http://info.energex.com.au/tracker/asp/lightningtracker.asp"
     onClick="window.open('http://info.energex.com.au/tracker/asp/lightningtracker.asp',
             'myWin', 'status, width=750, height=550'); return false">
     <U>http://info.energex.com.au/tracker/asp/lightningtracker.asp</U></a>
     <BR><BR><BR>
    </TD></TR>
  </TABLE></TD></TR>
</TABLE>

<!--*********************** WEBPAGE FOOTER *************************-->
<TABLE>
  <TR><TD colspan="3"> <DIV id="footer">
```

```
    Copyright &copy 2005 University of Southern Queensland. Design by Willy Ong.
    All right Reserved.<BR> Updated:25/10/2005 | Please read the
    <a href="Disclaimer.html" onClick="window.open('Disclaimer.html',
    'myWin', 'status, resizable=no, width=350, height=300'); return false">
    <U>Disclaimer</U></a> statement.<br>
  </DIV></TD></TR>
</TABLE>
</BODY>
</HTML>

<!--*********************************************************************************-->
<!--******************************** END of Index.html *****************************-->
<!--*********************************************************************************-->
```

B.2 `telescope.html` HTML Code

`telescope.html` is the webpage that provides live webcam for planetary photography and future link for ACP's web server for remote observing.

Listing B.2 `telescope.html`

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
      "http://www.w3.org/TR/html4/loose.dtd">
<HTML>

<HEAD>

<script language="javascript">
 <!--
//=========================================================================================
//================================= Start of JavaScript ================================


//******************************** Positioning function *********************************
// function for set footer position to suite display window
function setElementPos(navID, footerID) {
    //get pointer to navigation and footer object
    element1 = document.getElementById(navID);
    element2 = document.getElementById(footerID);
    //get size of current display window (web browser)
    var totalHeight =document.body.offsetHeight;
    var totalWidth =document.body.offsetWidth;
    //estimate width of footer and height of navigation bar
    totalWidth = totalWidth - 10 + 'px';
    totalHeight = totalHeight - 190 + 'px';
    //set the size properties for site navigation and footer
    element1.style.height = totalHeight;
    element2.style.width = totalWidth;
}

//*********************************** Initialisation ***********************************
function initialSetting() {
    //set footer position
    setElementPos('nav','footer');
    // start auto update for webcam image
    clockID = setTimeout('UpdateClock()', 2000);
}

//******************************** Auto Update functions ********************************
var clockID = 0;

function UpdateClock() {
   // clear timeout of previous setTimeout
   if(clockID) {
      clearTimeout(clockID);
      clockID  = 0;
   }
   //refresh webcam image
   refresh();
   // set timeout for recursive call of next iteration of update function
   clockID = setTimeout('UpdateClock()', 2000);  //refresh webcam image every 2 sec
}

function KillClock() {
   // clear timer
   if(clockID) {
      clearTimeout(clockID);
```

```
      clockID  = 0;
   }
}

//****************************** Image Refresh function *******************************
function refresh() {
    //set image source to latest modified image
    document.images.webcam.src='image/webcam.jpg?' + Date.parse(new Date().toString())
}

//================================= End of JavaScript ================================
//====================================================================================
//-->
</script>

  <!--set title, stylesheet, author, keyword for search, and description of document-->
  <TITLE>USQ Mt Kent Observatory - Telescope </TITLE>
  <LINK REL=stylesheet HREF="site.css" TITLE="Site Style Sheet" TYPE="text/css">
  <META NAME="author" CONTENT="Willy Ong">
  <META NAME="keywords"
        CONTENT="USQ astronomy observatory Toowoomba Mount Kent Australia">
  <META NAME="description"
        CONTENT="Main page for USQ observatory at Mt Kent">
</HEAD>

<!--************************************ HTML BODY **********************************-->
<!--set function call when loading and unloading document-->
<!--the auto positioning function not working IE when tested on student web-->
<!--but work well with Mozilla-->
<!--
<BODY onload="initialSetting()"
      onResize="setElementPos('nav','footer')"
      onunload="KillClock()">
-->
<BODY onload="initialSetting()"
      onunload="KillClock()">

<!--*********************** WEBPAGE HEADER **************************-->
<!--display header for Mt Kent Observatory Website-->
<TABLE ID="header" background="image/stripes.gif">
  <TR ID="header" >
    <TD ID="left" rowspan="2">
      <img ALT="USQ logo" src="image/usqlogo.gif"></TD>
    <TD ID="center"> <H1> Mount Kent Observatory </H1></TD>
    <TD ID="right" rowspan="2" background="image/trails_300px.jpg">
      <img src="image/stripes2.gif"></TD></TR>

  <!-- navigation for USQ home, usq search and contact us web pages-->
  <TR ID="header" >
    <TD id="link"> <a class="box" href="http://www.usq.edu.au">USQ Home</a> |
                   <a class="box" href="http://www.usq.edu.au/search">Search</a> |
                   <a class="box" href="contactus.html">Contact Us</a></TD></TR>
  <!--rule for seperating header and body-->
  <TR class="rule"><TD colspan="3"></TD></TR>
</TABLE>

<!--************************ WEBPAGE BODY ***************************-->
<TABLE>
  <TR class="space"></TR>
  <TR>
    <!--*********** site navigation *************-->
    <TD id="nav" rowspan="10"><UL id="site-nav">
        <LI><a href="index.html">Home</a></LI>
        <LI><a href="telescope.html">Telescope</a></LI>
        <LI><a href="gallery.html">Gallery</a></LI>
        <LI><a href="temp.html">Weather<BR> Station</a></LI>
    </UL></TD>
    <!--************* Main content **************-->
    <TD id="padding" colspan=2>
      <H4>O' Mara Telescope </H4><BR>
      O' Mara Telescope Remote Observing Project is still under progress.
      The equipment is as follow: <BR>
```

```
       - Telescope: Celestron CGE 1400, 14" aperture<BR>
       - Focuser: OPTEX celestron 3"<BR>
       - Robotic Mount: Paramoount ME<BR>
       - Camera: SBIG New Large Format<BR>
      <BR><BR><BR><BR>

      <H4> Webcam Imaging </H4><BR>
      Following is the webcam display for non-deep sky imaging:
      <TR><TD id="padding" colspan=2>
      <img alt="Webcam.jpg" id="webcam" src="image/webcam.jpg" onerror="KillClock()">
      <BR><BR></TD>
    </TR>
</TABLE>

<!--*********************** WEBPAGE FOOTER *************************-->
<TABLE>
  <TR><TD colspan="3"> <DIV id="footer">
    Copyright &copy 2005 University of Southern Queensland. Design by Willy Ong.
    All right Reserved.<BR> Updated:25/10/2005 | Please read the
    <a href="Disclaimer.html" onClick="window.open('Disclaimer.html',
    'myWin', 'status, resizable=no, width=350, height=300'); return false">
    <U>Disclaimer</U></a> statement.
    </DIV></TD>
  </TR>
</TABLE>

</BODY>
</HTML>

<!--*******************************************************************************-->
<!--****************************** END of telescope.html ***************************-->
<!--*******************************************************************************-->
```

B.3 `gallery.html` HTML Code

`gallery.html` is the gallery site for Mt Kent Observatory. Currently there are only two presented in this page.

Listing B.3 `gallery.html`

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
      "http://www.w3.org/TR/html4/loose.dtd">
<HTML>

<HEAD>
<script language="javascript">
 <!--
//======================================================================================
//================================== Start of JavaScript ===============================

function setElementPos(navID, footerID) {
    element1 = document.getElementById(navID);
    element2 = document.getElementById(footerID);
    var totalHeight =document.body.offsetHeight;
    var totalWidth =document.body.offsetWidth;
    totalWidth = totalWidth - 10 + 'px';
    totalHeight = totalHeight - 190 + 'px';
    element1.style.height = totalHeight;
    element2.style.width = totalWidth;
}

//================================== End of JavaScript =================================
//======================================================================================
//-->
</script>

  <TITLE>USQ Mt Kent Observatory - Gallery </TITLE>
  <LINK REL=stylesheet HREF="site.css" TITLE="Site Style Sheet" TYPE="text/css">
  <META NAME="author" CONTENT="Willy Ong">
  <META NAME="keywords"
        CONTENT="USQ astronomy observatory Toowoomba Mount Kent Australia">
  <META NAME="description"
        CONTENT="image gallery">
</HEAD>

<!--************************************ HTML BODY **********************************-->
<!--set function call when loading, resizing document-->
<!--
<BODY onLoad=setElementPos('nav','footer') onResize=setElementPos('nav','footer')>
-->
<BODY onLoad=setElementPos('nav','footer')>

<!--*********************** WEBPAGE HEADER **************************-->
<!--display header for Mt Kent Observatory Website-->

<TABLE ID="header" background="image/stripes.gif">
  <TR ID="header" >
    <TD ID="left" rowspan="2">
      <img ALT="USQ logo" src="image/usqlogo.gif"></TD>
    <TD ID="center"> <H1> Mount Kent Observatory </H1></TD>
    <TD ID="right" rowspan="2" background="image/trails_300px.jpg">
      <img src="image/stripes2.gif"></TD></TR>

  <!-- navigation for USQ home, usq search and contact us web pages-->
  <TR ID="header" >
    <TD id="link"> <a class="box" href="http://www.usq.edu.au">USQ Home</a> |
```

```
                   <a class="box" href="http://www.usq.edu.au/search">Search</a> |
                   <a class="box" href="contactus.html">Contact Us</a></TD></TR>
  <!--rule for seperating header and body-->
  <TR class="rule"><TD colspan="3"></TD></TR>
</TABLE>

<!--************************ WEBPAGE BODY ***************************-->
<TABLE>
  <TR class="space"></TR>
  <TR>
    <!--************* site navigation *************-->
    <TD id="nav" rowspan="10"><UL id="site-nav">
        <LI><a href="index.html">Home</a></LI>
        <LI><a href="telescope.html">Telescope</a></LI>
        <LI><a href="gallery.html">Gallery</a></LI>
        <LI><a href="temp.html">Weather<BR> Station</a></LI>
    </UL></TD>

    <!--***************** Title *******************-->
    <TD id="padding" colspan=2>
      <H4> Mt Kent Observatory Image Gallery </43><BR><BR>

    <!--********** ** Image Thumbnail *************-->
      <a href="image/image30.jpg" class="border"
         onClick="window.open('image/image30.jpg','myWin1',
         'status, menubar, scrollbars, resizable'); return false">
      <img class="border" alt="image30.jpg" src="image/image30thumb.jpg"></a>   

      <a href="image/image31.jpg" class="border"
         onClick="window.open('image/image31.jpg','myWin2',
         'status, menubar, scrollbars, resizable'); return false">
      <img alt="image31.jpg" src="image/image31thumb.jpg"></a></TD>
    </>
</TABLE>

<!--*********************** WEBPAGE FOOTER *************************-->
<TABLE>
  <TR><TD colspan="3"> <DIV id="footer">
    Copyright &copy 2005 University of Southern Queensland. Design by Willy Ong.
    All right Reserved.<BR> Updated:25/10/2005 | Please read the
    <a href="Disclaimer.html" onClick="window.open('Disclaimer.html',
    'myWin', 'status, resizable=no, width=350, height=300'); return false">
    <U>Disclaimer</U></a> statement.
    </DIV></TD>
  </TR>
</TABLE>

</BODY>
</HTML>

<!--*********************************************************************************-->
<!--****************************** END of gallery.html ******************************-->
<!--*********************************************************************************-->
```

B.4 `contactus.html` HTML Code

`contactus.html` is the HTML code for webpage which provide contact details.

Listing B.4 `contactus.html`

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
      "http://www.w3.org/TR/html4/loose.dtd">
<HTML>

<HEAD>
  <TITLE>USQ Mt Kent Observatory - Contact Us </TITLE>
  <LINK REL=stylesheet HREF="site.css" TITLE="Site Style Sheet" TYPE="text/css">
  <META NAME="author" CONTENT="Willy Ong">
  <META NAME="keywords"
        CONTENT="USQ astronomy observatory Toowoomba Mount Kent Australia">
  <META NAME="description"
        CONTENT="Contact Us">
</HEAD>

<!--*********************************** HTML BODY *********************************-->
<BODY>
<DIV id="spacer"></div> <!--content fill-->
<DIV id="contentwrap"><!-- contentwrap -->

<!--*********************** WEBPAGE HEADER **************************-->
<!--display header for Mt Kent Observatory Website-->

<TABLE ID="header" background="image/stripes.gif">
  <TR ID="header" >
    <TD ID="left" rowspan="2">
      <img ALT="USQ logo" src="image/usqlogo.gif"></TD>
    <TD ID="center"> <H1> Mount Kent Observatory </H1></TD>
    <TD ID="right" rowspan="2" background="image/trails_300px.jpg">
      <img src="image/stripes2.gif"></TD></TR>

  <!-- navigation for USQ home, usq search and contact us web pages-->
  <TR ID="header" >
    <TD id="link"> <a class="box" href="http://www.usq.edu.au">USQ Home</a> |
                   <a class="box" href="http://www.usq.edu.au/search">Search</a> |
                   <a class="box" href="contactus.html">Contact Us</a></TD></TR>
  <!--rule for seperating header and body-->
  <TR class="rule"><TD colspan="3"></TD></TR>
</TABLE>

<!--************************ WEBPAGE BODY **************************-->
<TABLE>
  <TR class="space"></TR>
  <TR><TD id="padding" align="center" colspan=3>
       <BR>
       <H3> Contact Us </H3><BR><BR>
  </TD></TR>
  <TR><TD id="indent" colspan=3>
       <H4> USQ - Faculty of Science (Astronomy) </H3><BR><BR>
       <b>Email:</b>   
         <a href="mailto:astronomy@usq.edu.au"><U>astronomy@usq.edu.au</U></a>
         <BR><BR>
       <b>Phone:</b>   
         07 - 46312094<BR><BR>
       <b>Website:</b> 
         <a href="www.usq.edu.au/astronomy"><U>www.usq.edu.au/astronomy</U></a>
       <BR><BR>
  </TD></TR>
 </TABLE>
```

```
</DIV><!-- contentwrap -->

<!--*********************** WEBPAGE FOOTER *************************-->
<DIV id="footer2">
  Copyright &copy 2005 University of Southern Queensland. Design by Willy Ong.
  All right Reserved.<BR> Updated:25/10/2005 | Please read the
  <a href="Disclaimer.html" onClick="window.open('Disclaimer.html',
  'myWin', 'status, resizable=no, width=350, height=300'); return false">
  <U>Disclaimer</U></a> statement.
</DIV>
</BODY>
</HTML>

<!--*********************************************************************************-->
<!--****************************** END of contactus.html ****************************-->
<!--*********************************************************************************-->
```

B.5 `Disclaimer.html` HTML Code

`Disclaimer.html` is the webpage for disclaimer notice for the website.

Listing B.5 `Disclaimer.html`

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
      "http://www.w3.org/TR/html4/loose.dtd">
<HTML>

<HEAD>
  <TITLE>Disclaimer</TITLE>
  <LINK REL=stylesheet HREF="site.css" TITLE="Site Style Sheet" TYPE="text/css">
  <META NAME="author" CONTENT="Willy Ong">
  <META NAME="description"
        CONTENT="USQ Mt Kent Observatory Disclaimer">
</HEAD>

<!--************************************ HTML BODY **********************************-->
<BODY>
<H5><BR>
This website is provided on an "as is" basis and therefore, makes no warranties
of any kind, expresse, or implied to the maximimum extent permitted by applicable
law.<BR><BR>
By using this website, you agree to accept all risks and responsibilities for
losses, damages, costs, and other consequences resulting directly or indirectly
from using it and any information or material available on it.<BR><BR>
THe Rain Radar Map and Satellite Imagery are copyright of Bureau of Meteorology
Australia. Please note their <a href="http://www.bom.gov.au/copyright.shtml">
<U>copyright</U></a> notice.
<H5>
</BODY>
</HTML>

<!--*********************************************************************************-->
<!--***************************** END of Disclaimer.html ***************************-->
<!--*********************************************************************************-->
```

B.6 `site.css` Stylesheet for website

`Site.css` is the stylesheet for all of the webpages.

Listing B.6 `site.css`

```
/*****************************************************************************************
** Mt Kent Observatory page style sheet.                                               **
** Date:    5 Feb, 2004                                                                **
** Authors: Willy Ong                                                                  **
**                                                                                     **
*****************************************************************************************/

html{
       height: 100%;     /* positioning purpose */
}

body {
       color: #cccccc    !important;  /* font colour - light grey */
       font-family: Courier New, Rockwell, verdana, Arial;
       background-color: #000000;
       margin: 0;
       padding: 0;
       border: 0;
       height:100%    /* positioning purpose */
       }

/********************************** heading style ************************************/
/* The font size is set to suit different browser */
/* font colour is light grey to suit the black background */
h1 {
    font-family: Courier New, Rockwell, verdana, Arial;
    color: #cccccc;
    font-style: normal;
    font-weight: bold;
    font-size: xx-large;
    margin:0
   }
h3 {
    font-family: Courier New, Rockwell, verdana, Arial;
    color: #cccccc;
    font-style: normal;
    font-weight: bold;
    font-size: x-large;
    margin:0
   }

h4 {
    font-family: Courier New, Rockwell, verdana, Arial;
    font-style: normal;
    font-weight: bold;
    font-size: large;
    margin:0
   }

h5 {
    font-family: Arial, Courier New, Rockwell, verdana;
    font-style: normal;
    font-weight: normal;
    font-size: small;
    Padding:10px 10px 10px 10px
   }

/********************************** webpage header style ******************************/
```

```
/* The table rows at the top of the page that contain the headers */
table#header {
              border-style: hidden;
              border-spacing: 0px;
              border-collapse: collapse;
              padding: 0px;
              width: 100%
             }

/* The cell on the left containing the USQ image */
tr#header td#left {
                   padding:        0;
                   vertical-align: middle;
                   text-align:     left;
                   height:         72px;
                   width:         121px
                   }

/* Cell on the right containing the Mount Kent image */
tr#header  td#right {
                     background: "image/stripes.gif";
                     padding: 0;
                     width:   300px;
                     vertical-align: top
                     }

/* Cell in the center that contains the heading text */
tr#header td#center {
                     padding:    0;
                     height:     98px;
                     text-align: center
                    }

/* general link below the heading text */
tr#header td#link {
                   padding:      0;
                   text-align:   right;
                   height:       20px;
                   font-size:    x-small
                  }

/************************************ rule and space *********************************/
/* The horizontal rule that separates the header from the body */
tr.rule {
         margin:  0;
         padding: 0;
         background-color: #aaaaaa;
         height:  5px
        }
/* The standard space around the body components is 10 pixels
   This is the space between the rule and the ToC and the Contents
   of the page */

tr.space {
         margin:  0;
         padding: 0;
         height: 10px
        }

/************************************* Anchor style **********************************/
/* Standard Anchors*/
a:link {
        color:           #9999ff;
        text-decoration: none
       }

a:visited {
           color:           #9999ff;
           text-decoration: none
          }

a:active {
```

```
          color:            #ffffff;
          background-color: #bcc8d9;
          text-decoration:  none
         }

a:hover {
         color:            #ffffff;
         text-decoration:  none
}

/* Anchors not in box */
a.box:hover {
             background-color: #9999ff;     /* Background colour when the cursor
                                               hovers over the link */
             color:            #ffffff;
             text-decoration:  none
            }

/************************************ border style **********************************/
img.border {border:solid 2px; color:#9999FF;}
a.border {
          border: 2px solid  #9999ff;
          background-color:  #9999FF;
          color:             #9999FF;
          text-decoration:   none
          }

/**************************************** DIV style ************************************/
/* spacer to fill up display window for footer setting */
div#spacer{
           display:   block;
           height:    94%;
           float:     left;
           width:     1px;    /* non zero value to enable the spacer height setting  */
           font-size: 1px     /* make sure width is not overridden */
          }

div#contentwrap{
                display: block;
                width:   99%       /* to suit but leave room for spacer */
               }

/* This is the MAIN Division that contains the contents of the page */
div#content {
             background-color: #ddebff;   /* background colour */
             padding:          10px;
             border: 2px solid #002151;   /* Border width style and colour */
             margin: 0px 10px 0px 0px     /* 10 pixel gap between the border and
                                             contents of the DIV */
        }

/* This is the IMAGE Division that contain overlay images */
DIV.initial{
            position: absolute;              /* absolute positioning for layering */
            top:      0px;
            z-index:  2;                     /* set higher layer than usual image */
            visibility:hidden;               /* hide initial overlay image */
           }

/* This is the FOOTER division at the bottom of the page */
div#footer {
       height:      55px
       color:       #888888;
       padding-top: 5px;
       text-align:  center;
       font-family: Arial, Courier New, Rockwell, verdana;
       font-size:   x-small;
       border-top:  solid #888888 1px
       }

/* This is the FOOTER division for webpage with no navigation column*/
div#footer2 {
```

```
       display:     block;
       clear:       both;
       height:      4%
       color:       #888888;
       padding-top: 5px;
       text-align:  center;
       font-family: Arial, Courier New, Rockwell, verdana;
       font-size:   x-small;
       border-top:  solid #888888 1px
       }

/********************************** navigation style ********************************/
/* the UL LI elements to only those elements that appear in navigation column*/

/* The main unordered list */
ul#site-nav {
             margin: 0px 10px;
             padding: 0;
             list-style: none
            }

/* The main list items */
ul#site-nav li {
                border: 0px none;
                border-bottom: 2px solid #000000
               }

/* The style of the anchors used with main list items */
ul#site-nav li a {
                  background-color: #444466;
                  color:            #ffffff;
                  display:          block;
                  margin:           1px;
                  border:           0px;
                  padding:          2px 2px 4px 4px;
                  text-align:       left;
                  font-size:        small;
                  text-decoration:  none
                 }

ul#site-nav li a:active {
                         background-color: #003399;
                         color: #ffffff;
                         text-decoration: none
                        }

ul#site-nav li a:hover {
                        background-color: #9999ff;
                        color: #ffffff;
                        text-decoration: none
                        }

/************************************** table style ***********************************/
/* the table for displaying

/* the table contains RAIN RADAR MAP */
table#RainRadar {
                 border-style:    hidden;
                 border-spacing:  0px;
                 border-collapse: collapse;   /* Needed for IE */
                 padding:         0px;
                 width:           524px;
                 text-align:      left;
                 font-size:       small
                }

/* the table contains SATELLITE IMAGERY */
table#Satellite {
                 border-style:    hidden;
                 border-spacing:  0px;
                 border-collapse: collapse;   /* Needed for IE */
                 padding:         0px;
```

```
                 width:          640px;
                 text-align:     left;
                 font-size:      small
                }

/* the cell for NAVIGATION list*/
td#nav {
        vertical-align: top;
        border-right:   solid #aaaabb 2px
       }

/* the cell for CONTENT */
td#padding{
           padding:        0px 0px 0px 20px; /*padding top bottom right left*/
           vertical-align: top;
           font-size:      small
}

/* the cell for indented CONTENT */
td#indent {
           padding-left:   50px;
           font-size:      small;
}

/************************************ printing style ********************************/

/* Modification of basic FORM elements */

input.reset, input.submit  { background:   #758faf;
                             border-color: #758faf;
                             font-weight:  bold;}

input.text, textarea { background:   #f4f8fe;
                       border-color: #ddebff;
                       font-family: Courier New, Rockwell, verdana, Arial;
                       font-style: normal;
                       }

@media print {
div#nav-links { display: none; }
tr#header  td#right img { display: none; }
tr#header  td#left  img { display: none; }
tr.rule { background-color: #000000; }
body{
     background-color: #ffffff;
     font-color:#000000;
     border: 1px solid #000000;
            }
a:link { color: #000000; }
a:visited { color: #000000; }
}

/************************************************************************************/
/************************************ End of site.css *******************************/
/************************************************************************************/
```

Appendix C

Permission Letter

Australian Government

Attorney-General's Department

Information Law and
Human Rights Division

21 September 2005

Mr Willy Ong
University of Southern Queensland
5/10 Buckand Street
TOOWOOMBA QLD 4350

Copyright Request - Reference Number - 10573

Dear Mr Ong

I refer to your request of 21 September 2005 in which you seek permission to reproduce/ communicate and/or adapt the following Commonwealth of Australia copyright material:

- Marburg Rain Radar and Satellite image

for inclusion in your research project.

Copyright Permission

Permission is granted to reproduce the advised material for the specific purpose requested on a revocable, non-exclusive, non-transferable basis without charge subject to the following terms:

(a) The material must be used in an appropriate context and reproduced accurately without distortion of meaning.

(b) The source of the material must be recognised through the inclusion of an acknowledgment.

The acknowledgment must state the full title of the source, the author or author body, publisher and date of publication, if applicable, followed by the words 'copyright Commonwealth of Australia reproduced by permission'.

(c) The material must not be used for commercial sale or profit.

Separate permission is required for commercial use

Please note that this permission does not apply to any illustration, diagram or text over which the Commonwealth does not hold copyright, but which may be part of or contained within the material specified above. Please examine the material carefully for evidence of other copyright holders. Where a copyright holder, other than the Commonwealth, is identified with respect to a specific item in the material that you wish to reproduce, please contact that copyright holder directly.

Yours sincerely

Dorothy Sweeney
Commonwealth Copyright Administration
Copyright Law Branch

Telephone – 6250 6200
Website – http://www.ag.gov.au/cca

Appendix D

Screenshots

D.1 `index.html` Screenshots

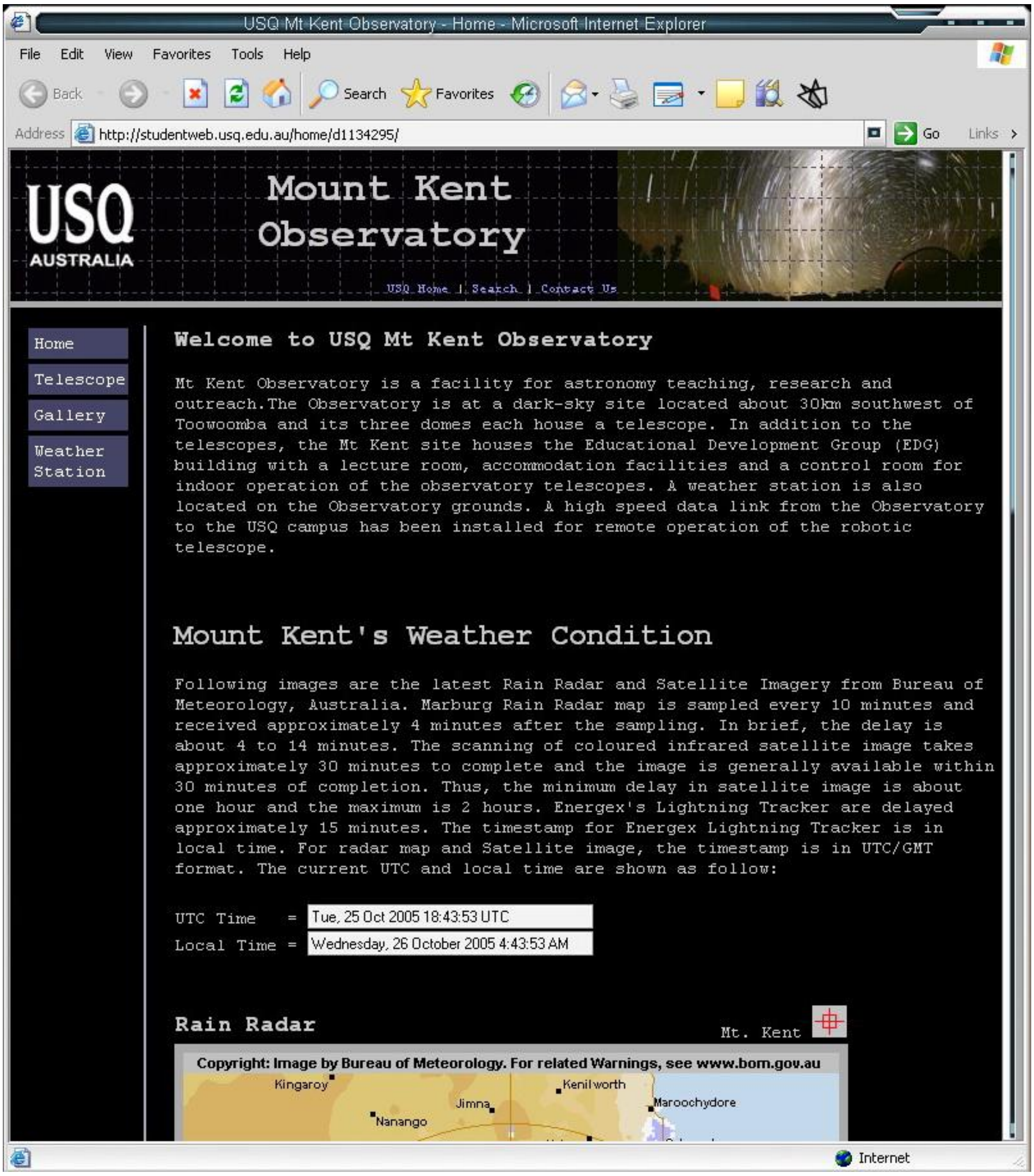

Figure D. 1a: Top page of *index.html* opened with IE6

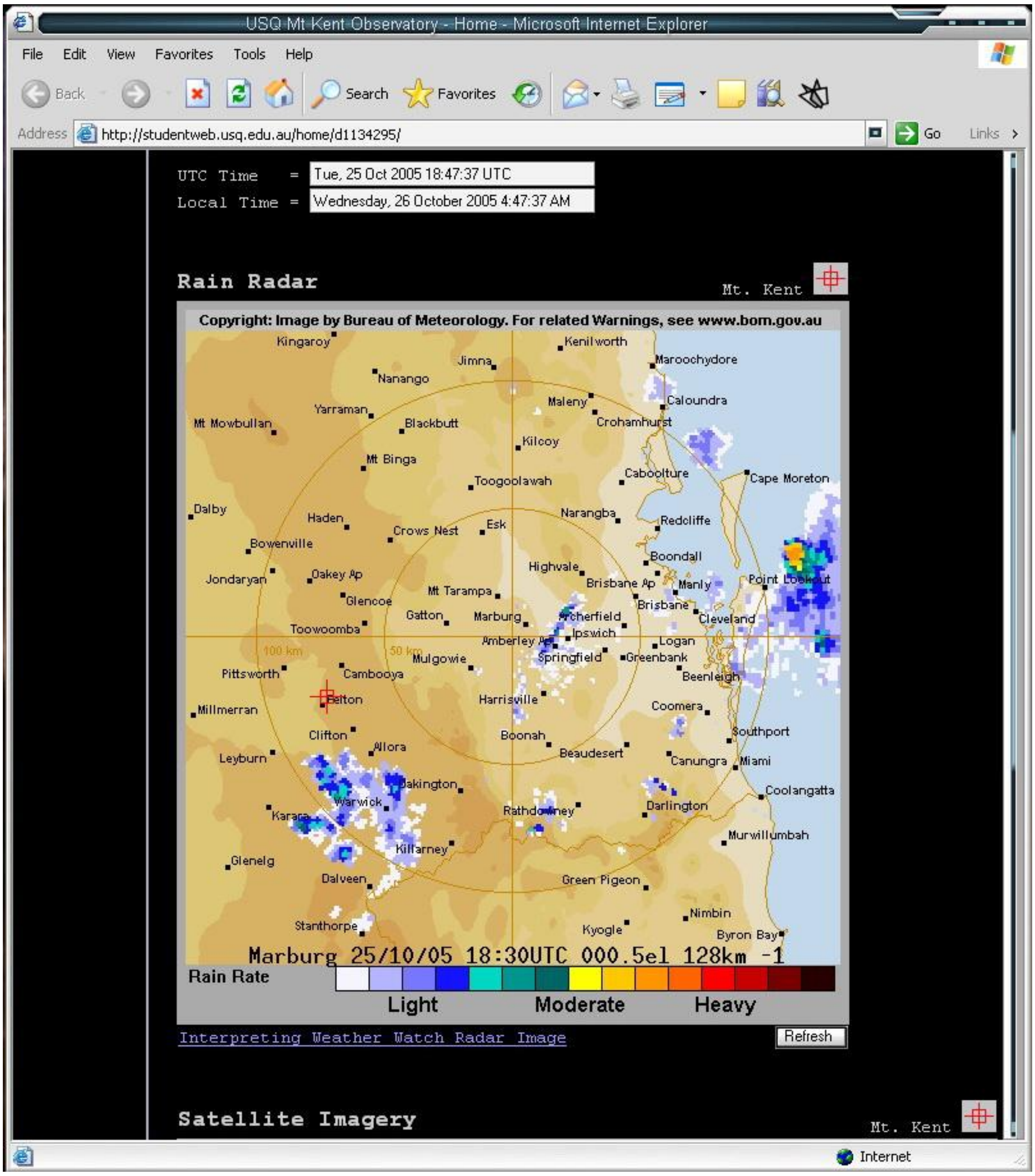

Figure D. 1b: Middle page of *index.html* opened with IE6

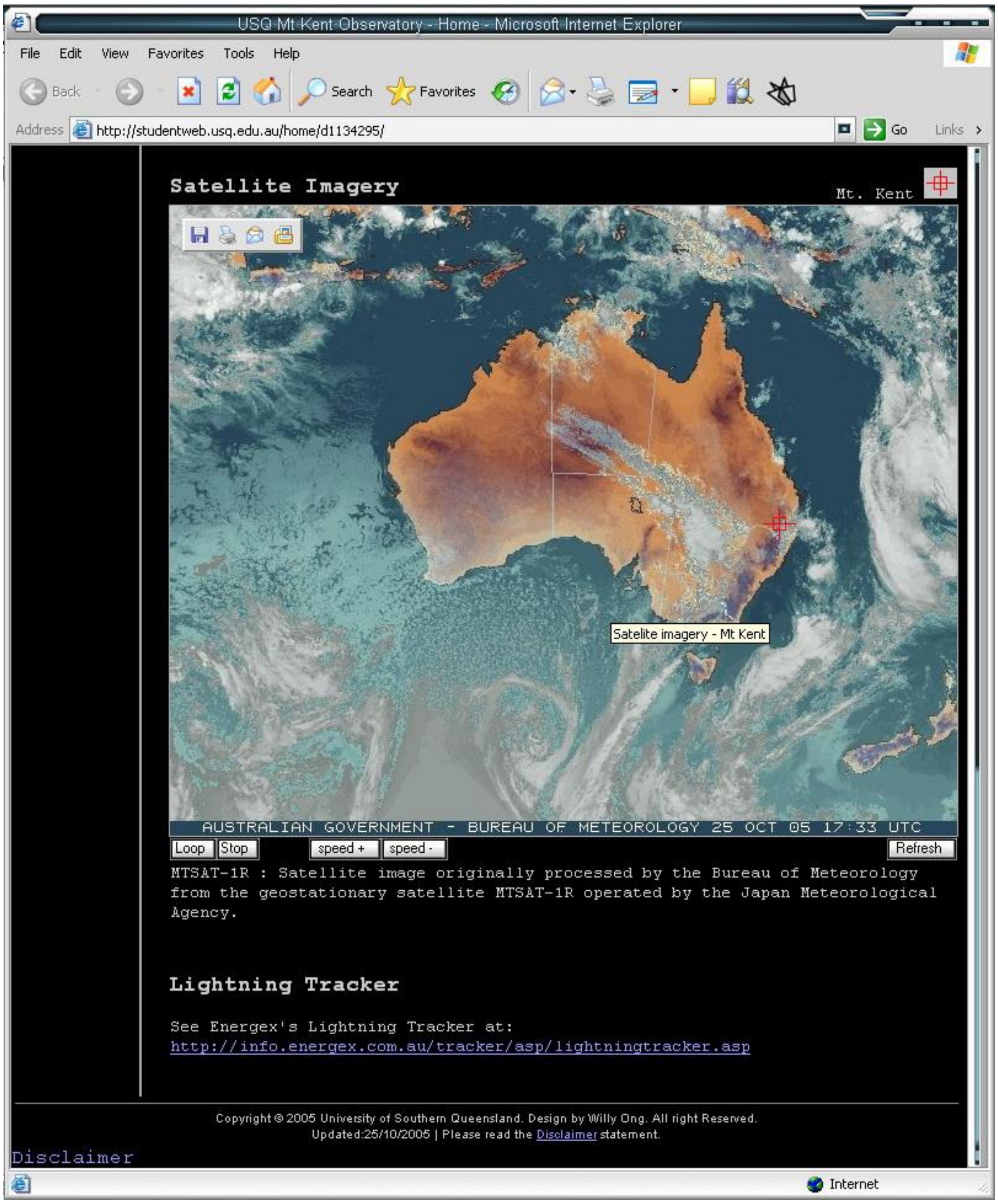

Figure D. 1c: Bottom page of index.html opened with IE6. The disclaimer notice on left down corner is USQ disclaimer notice which automatically attached to the web page.

D.2 `telescope.html` Screenshot

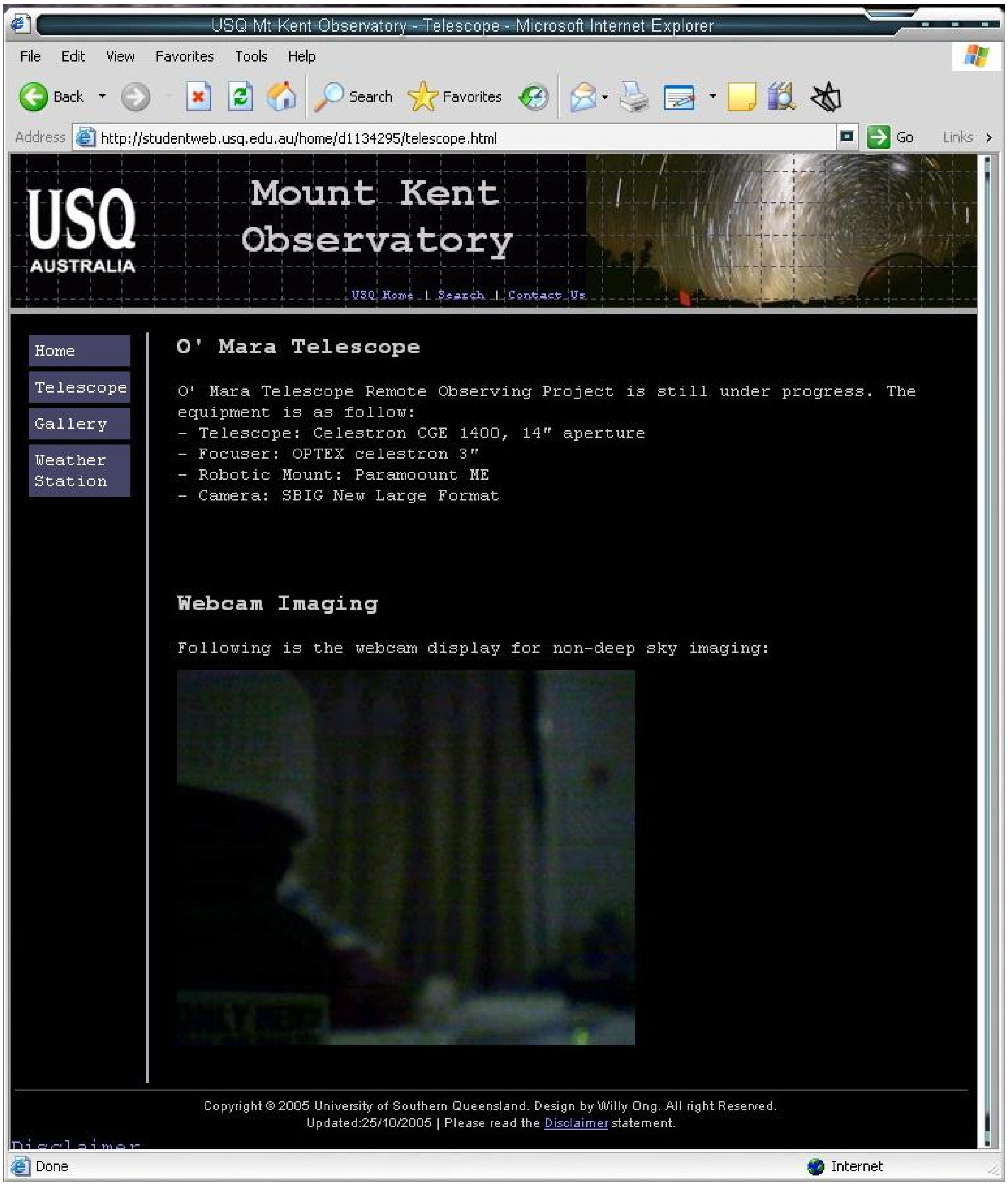

Figure D. 2: *telescope.html* opened with IE6. With addition of USQ Disclaimer, the page will always few pixels wider.

D.3 `gallery.html` Screenshot

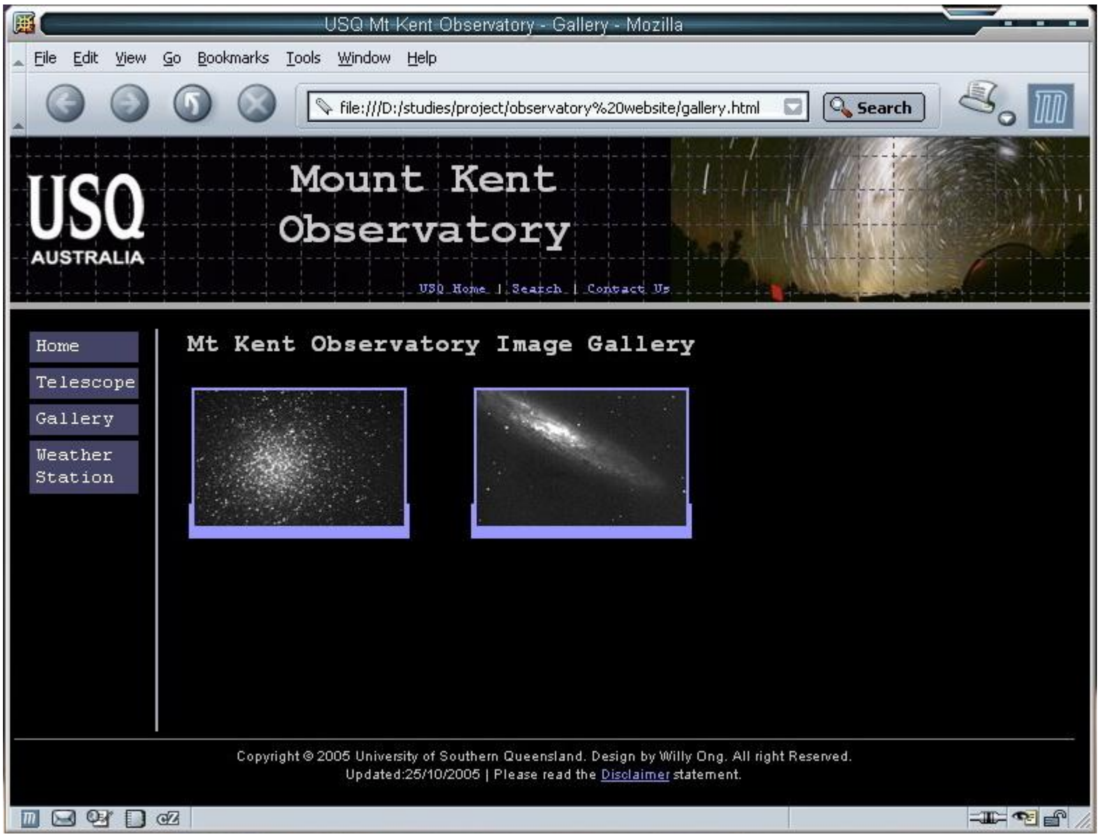

Figure D. 3: *gallery.html* opened with Mozilla1.7.12. This was tested on the PC itself. The footer are repositioned to fit the page

D.4 `contactus.html` Screenshot

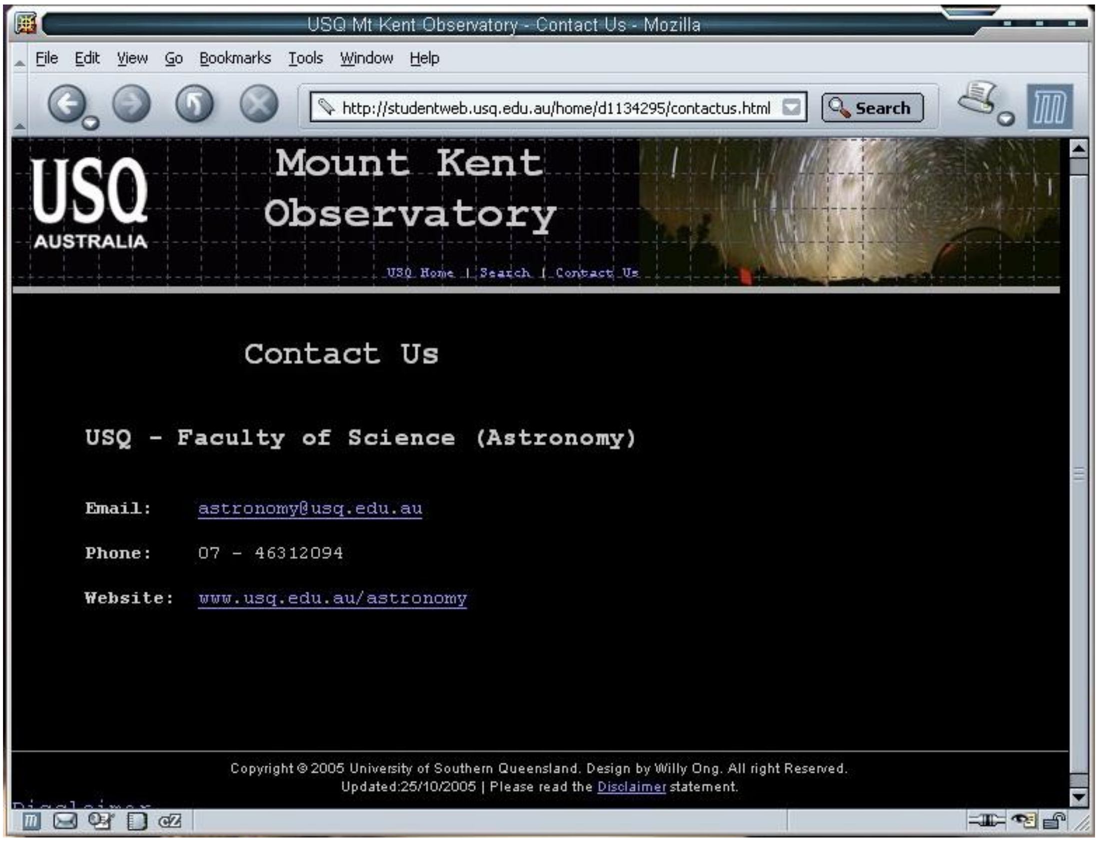

Figure D. 4: *contactus.html* opened with Mozilla 1.7.12

D.5 `Disclaimer.html` Screenshots

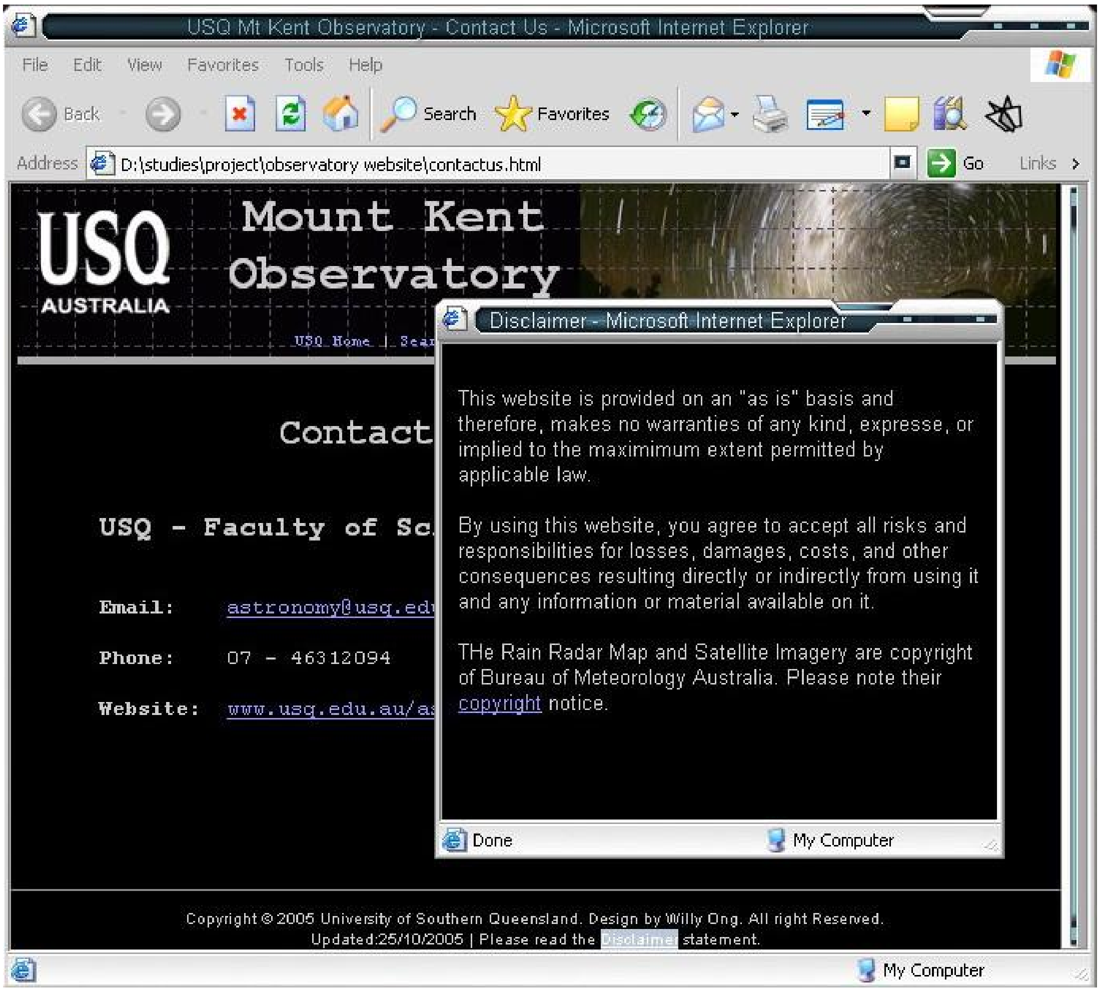

Figure D. 5: Popup window of *Disclaimer.html* with contactus.html at back

D.6 `temp.html` Screenshot

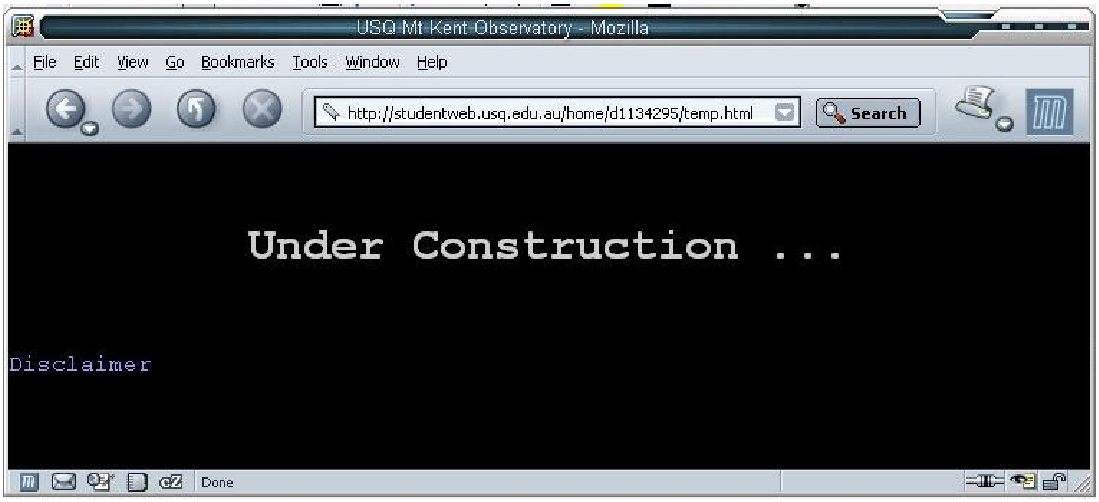

Figure D. 6: temporary page opened with Mozilla1.7.12.