**CE101 Team Report Assignment**

**Team:** *(Team H)*

**Team Leader:** *Emmanuel Odame-Asante*

**Project Manager:** *Ted Fey*

**Team Specialists:** *Simas Kairys, Jeffrey Chujor, Benjamin Higgs, Desmond Wei Chua*

**Sections completed:**

### Emmanuel Odame-Asante (Team Leader)

### Executive Summary

### Introduction to Team Learning

### Report of tasks done.

### Conclusion

### Ted Fey (Team Project)

### Project Management section.

* Report of tasks done.

### Simas Kairys

### Product Design

* Product Implementation
* Report of tasks done.
* Product specification

### Benjamin Higgs

* Report of tasks done.
* Product Specification
* Product Design
* Product Implementation

### Jeffery Chujor

### Legal Matters

### Ethical matters

### Health and Safety matters

**Desmond Wei Yan Chua**

* Report of tasks done.
* Product Testing

### 

Table of Contents

Chapter 1 The Executive Summary (508 words) 5

Chapter 2 Team Working (840 words) 6

2.I An introduction to Team Working 6

2.II Team Activity Report 8

2.II.a The team effort summary table 8

2.II.b Detailed report of each team members contribution to the project 8

Chapter 3 Product Development (3901 words) 12

3.I An introduction to Product Development 12

3.II The Team Product 12

3.II.a The product specification 12

3.II.b The product design 13

3.II.c The product implementation 15

3.II.d The product testing 16

3.III Context 18

3.III.a Legal matters 18

3.III.b Ethical matters 18

3.III.c Health & safety matters 18

Chapter 4: Project Management (962 words) 21

4.I An introduction to Project Management 21

4.II Project Management Report 22

4.II.a A description of the Gantt chart 22

4.II.b An evaluation of the project management 22

Chapter 5: Conclusions (532 words) 24

Appendix 25

A. Python Code 27

B Team effort summary table 36

C Project management Gantt chart 38

# Chapter 1 The Executive Summary (508 words)

# After months of developing and planning Team H finally produced our final product called Hotel Scrapper. This team report explicitly states the process of planning, developing and testing of our product. This does not only show the progression of our product but also shows Team H developing as a team. Although we communicated at the start of the project it is not as efficient as now were we communicating regularly on Team Forum and primarily on whatsapp. This Team Report was divided by all our team. Two Product Specialist Simas and Ben wrote the section about the Report as they spent the most time developing the product and know about the actual program the most with me assisting with the Product Specification. The Project Management section was obviously taken on by our PM Ted and we had another product specialist handled the section Context. Throughout the Team Report multiple references are being made. As a team we made multiple references to lectures and materials on Moodle which helped us in the project. There are also a few references from online books on team developing which helped us as a team. In depth description of that is in the Team Working stage. Team working section used references from all sources such as quotes and explanation from Mr. Vickers lectures and extra notes and online books about success in Team Management. It is stressed that a team has to have exquisite communication between team members, as a team our communication has improved dramatically. Lessons on communication were primarily taught to us by Mr. Vickers in the first few lectures. The team report then goes on the report of all members of the team. In our team of six we all contributed, it is fair to say two of our Team specialists who actually developed the product invested more time than others in the team but no one in the team did nothing. Everyone contributed some time. Again, this section is heavily referenced with links to our team forum and also our Team database and précis section. After this section there is the portion about the actual Product. As stated before this was primarily handled by the two team specialist who had the most knowledge about the product. Our product is quite sophisticated and there are portions of the program that is beyond our scope for the little Python language we acquired from the start of the year. However one of the project specialists is very strong with Python from doing it before university so the coding was majorly handed by him, hence him writing a lot of the section on the Product. However portions such as Product Specification and Testing were done by other members in the team. The conclusion of the Team Report was handled by myself which obviously concludes the whole document and project. In the Conclusion section I mention how the team and product have developed and if I were to go back at the beginning of the year what we should of changed as a team.

# 

# Chapter 2 Team Working (840 words)

## 2.I An introduction to Team Working

What is Team Working? There are various definitions of what Team working actual is. Such as “work done by several associates with each doing a part but all subordinating personal prominence to the efficiency of the whole”, [1] However this definition can be interpreted in a few different ways. A more complete definition of Team Working is “When a group of people work together cohesively, towards a common goal, creating a positive working atmosphere and supporting each other to combine individual strengths to enhance team performance”[2]. Even though this definition is an accurate description of Team Working, most definitions have the same theme. They all say a team (group of people) wanting to work together to achieve something. This Team Working philosophy is applied to our project. We are a group of individuals studying Computer Science making a program for client Antony Vickers. Obviously it’s important for the team to be efficient. In lectures it was shown how teams can fail. These are dysfunction teams [3]. It is said that five areas of dysfunction are: “Absence of trust, Fear of Conflict, Lack of Commitment, Avoidance of accountability and Inattention to results”. [4] With these characteristics in the team it can hinders the team goal and which can cause the final product to suffer. This were specific team roles are introduced. Team Leaders are responsible making a team not reach that stage and ensure development goes smoothly. Like Anthony Vickers said, “The Team Leader is there to keep the team moving along”. [5] If a team is desperately failing and the Team Leader does not rise to prevail this could make the team completely fail. Team Leaders are most likely to have certain skills such which make a prominent Team leader. Certain skills needed are that they need to be mature and confident. [6] It is also stated that “Effective team leaders communicate clearly”. [7] Furthermore it is also stated that “Effective team leaders possess exceptional organizational skills”. [8] Those are a few very key characteristics that an effective Team Leader will possess. There also more desirable traits; “A quality team leader treats team members fairly”. [9] And finally “An effective team leader is honest and open with his team members.” [10]. A Team with an exceptional team leader is a very huge advantage for the development process. For our project, questionnaires were done to determine roles in the team. I had higher scores on my communication traits so that made me volunteer to be a Team leader. In addition to a Team Leader another important role in this Project is the project manager. It’s also important for the Team leader to actually learn their team and understand the strength and weakness of individuals. For example for this project I assigned tasks accordingly to specific team specialists’ strengths. So an individual that is good at Python coding language will get more coding exercises than writing. Like the Team Leader, Project Manager has to have exquisite communication skills. Anthony Vickers also stated that “Project manager should be someone who is organized and good at time management.”[11] This is important as they are responsible for important documents such as project plans and Gantt charts. Team specialists are individuals that will “take on particular specialist aspects of the Product development”. [12] It is important that the Team Leader assigns each individual roles that suites their strengths. As a team, communication is one of the most important aspects in an efficient team. It is stated that “Team communication skills are critical for ensuring the success of the team effort, whether the team is charged with creating a new product, making a process improvement or planning a summer picnic”. [13] This is reflected in our team project were our team communicates on a regular basis. We have a highly active whatsapp group and regularly meet up to discuss plans. We are together every fortnight for the laboratory sessions and also meet up regularly at level up or at the lime room if we need to bring anything up about the project. This ensures that we are always communicating with each other and sharing ideas. Another important aspect of team work understands what every individual strengths and what they like to do. Week one the whole team always communicates regularly about tasks and what needs to be done to get the best out of the team.

**References**

**[1]** Merriam-Webster Dictionary online. <http://www.merriam-webster.com/dictionary/teamwork>

**[2]** Online source- The Happy Manager. <http://www.the-happy-manager.com/articles/what-is-teamwork/>

**[3]** Antony Vickers Lecture 2 notes. <https://moodle.essex.ac.uk/mod/book/view.php?id=208150&chapterid=1604>

**[4]** Patick Lencioni “The five dysfunctions of a team” pg.1

**[5]**Anthony Vickers Week 3 notes. <https://moodle.essex.ac.uk/mod/book/view.php?id=208150&chapterid=1605>

**[6]**Anthony Vickers Week 3 notes. <https://moodle.essex.ac.uk/mod/book/view.php?id=208150&chapterid=1605>

**[7]** Online source- “The 10 Effective Qualities of a Team Leader. <http://smallbusiness.chron.com/10-effective-qualities-team-leader-23281.html>

**[8]** Online source- “The 10 Effective Qualities of a Team Leader. <http://smallbusiness.chron.com/10-effective-qualities-team-leader-23281.html>

**[9]** Online source-“The 10 Effective Qualities of a Team Leader. <http://smallbusiness.chron.com/10-effective-qualities-team-leader-23281.html>

**[10]** Online source-“The 10 Effective Qualities of a Team Leader. <http://smallbusiness.chron.com/10-effective-qualities-team-leader-23281.html>

**[11]** Anthony Vickers Week 3 notes. <https://moodle.essex.ac.uk/mod/book/view.php?id=208150&chapterid=1605>

**[12]** Anthony Vickers Week 3 notes. <https://moodle.essex.ac.uk/mod/book/view.php?id=208150&chapterid=1605>

**[13]** Importance of Team Communication Skills. Leigh Richards. http://smallbusiness.chron.com/importance-team-communication-skills-3079.html

## 2.II Team Activity Report

### 2.II.a The team effort summary table (see Appendix E)

### 2.II.b Detailed report of each team members contribution to the project

Emmanuel: My role in this project is Team Leader. My main responsibilities are managing and organizing the team. This includes arranging and planning meetings aside from our laboratory sessions every fortnight, setting specific tasks to members of the team and also motivating the team. I created a whatsapp group with only our team members to ensure optimum communication. On the Team forum I will post important updates of deadlines that has to be done. Also made sure that deadlines are set and tasks are always completed. In addition to managing the team I have various tasks such as initializing the Gantt chart to send to the project manager. On the early researching stage of the project I researched similar websites designs for ideas and compiling images together in a word document on what our layout could look like, (Evidence of this is on the reference database page). I also researched about the main framework which we used in our code called ‘Beautiful Soup and linked instructional videos and information on the team forum that can help the team specialists with the development of the product. I also completed to IEEE like précis’. One of the documents outlined and described what an IEEE/SRS specification is and I created my part of the SRS in relation to this project. (Also on reference database page). In addition to this I helped the creation of the website we used by creating filler information that is used for the website. This was uploaded onto the team forum for the team specialists to see. I also helped with the development of the team report. In addition to writing the Executive Summary and Introduction to Team working, I completed Conclusion section and helping finalize the product specification section. I created a presentation for the rest of the team about IEEE specification and about SRS which specified the main sections of them both to a good standard.

Ted: As project manager my main responsibilities were maintaining the team logbook and Gantt chart throughout the project. I started the logbook from week 5 of the course and we have meetings at least once a fortnight, every one of these meetings has an agenda, the minutes of the meeting are recorded and any tasks that have been set to team members are listed. I also used the logbook to keep track of everyone’s attendance to meetings and it is shared with the team so if someone misses a meeting they can have a look at the minutes to find out what was discussed. Unfortunately I missed a couple of meetings myself so the other team members wrote the minutes down and sent them to me so I could add them into the logbook. Emmanuel created the initial Gantt chart file and sent it to me so I could work on it. At first I broke it up into sections, then I broke it up further into a week-by-week chart so it’s easy to see what each team member was doing each week. I uploaded some pictures of other comparison websites as references so we could gain some inspiration for the design aspect of the project. I made a précis of the second section of the IEEE standards so it was quicker to read for the rest of the group. Finally I did a presentation to the rest of the group about the team report, going through the layout and content so we all had a better understanding of how to write it; I also reviewed the other team member’s presentations.

Simas: I selected a team specialist role as I really like coding and I think I posses proper amount of knowledge in this field. I was extremely happy when the team chose my idea - to create a review scrapping website. From that moment I wasn't able to stop thinking about the design and implementation of our project. Within 7-8 week I came up with the system design, did many presentations in the team meetings, encourage discussion about it and uploaded descriptions, functional diagrams to the team forum. After this we divided a project in two parts, I took over Python part and Ben took over PHP part.

I wrote a précis about BeautifulSoup and uploaded it to the team forum, so my team mates would have better understanding how this module works. I also did a presentation about Python modules which are used in our project. I did a in-depth research about the review pages we are going to scan(how they display their pages, how the pagination works, do they have any human verification methods installed and so on.) With this information I was able to develop the Python program which access desired website, downloads available pages, extracts reviews and records them to a database . I also designed and implemented the beta database for the project. I spent many hours on writing the code, testing, configuring modules and adjusting the program flow. I also did the basic testing of the program. It was a great challenge and I have really enjoyed it. The development wasn't easy because we are scrapping data from a dynamic page, the system has to do many checks to make sure that correct data is imported.

When we reached the point when both of the project parts were working properly, me and Ben joined our forces and started the final step of our project development – merging everything together. We had many meetings so we were able to discuss the problems we are facing and decide how to solve them. I configured VPS server and we finalized the project by transferring it from the local machines to our new server. After setting up our server, both of us worked very intensively to synchronize PHP and Python parts, establish a proper communication between those two entities. We did a lot of system debugging, so the program would work smoothly and the fundamental mistakes would be eliminated. This allowed me to find and fix a major bug in a python script(a lot of reviews got ignored because of the different encoding). I wrote a précis which explained how our server could be accessed and introduced the basic management programs. Finally, Ben and me wrote a third chapter for our team report as we had the best understanding how the system works and is designed.

Jeffrey: As a product specialist, due to my weaknesses in Python I opted for the research route in order to help my group develop a product that made sense. My research involved detailed analysis of how hotel review websites would work, this included what the ratings were based on, how the customers would comment, what the stars would mean for the individual hotels and etc. From this, , my team were able to grasp an understanding of how we would be able to use the basics of hotel review websites and transfer it to our unique idea of a ‘meta-analysis of hotel review sites’. Although Python isn’t my strongest point, I still carried out extensive research into the ‘Beautiful Soup’ function that would help us out with the coding aspect of our project (extracting the data we needed from the hotel review sites on the internet) . In my PowerPoint presentation given in week 19, I provided examples of code that could be used in our program (further explained in depth by Simas in another presentation). Also, in the same presentation I went over the different methods (design, problem solving) that we were taught in the lectures and applied it to how we used them as a group to develop our project.

Ben: Being a specialist for the team, my job involved the coding and creation of the website side of the system that we have created. This is something that I have invested a lot of my time doing, and have spent countless late nights finishing functions and pages. We started by planning our system, and we all looked into the possibility of scraping hotel review websites. Once we had made a choice as a group, I summarised a group of possible extensions that we could use in python to help physically get the data from the websites. I collated these summaries and uploaded them to the forums and discussed these with Simas, the other specialist for coding in the group. The plan to use a PHP based website was something that we discussed from an early stage. I recommended myself to do this because I already had plenty of experience creating similar systems using this language. I started the design of the website early on, with some sketches and basic web page templates based on the framework that we would be using, Bootstrap 3. Once we had discussed the idea of using this framework for our site, I set about learning and summarising all of the pros and cons of the system. This I also uploaded to the team forum. Once I felt that I had a good understanding, I set about creating some basic pages that formed the basis of our system. I then created a presentation as a summary of my understanding into the framework and how it would benefit us as a group and how it would influence out project and the utilities that we could use from it, such as interactive graphs and charts. After drawing up a draft product specification, I began to code the systems main functions and outputs based on a draft database schema for the system. These took a lot of time due to the actual function creation, but also the validation of inputs and error displaying on the website. These were debugged by myself on a local server to make sure that the system would be functional. After this, I joined forces with Simas, and after countless one on one meetings we combined the code and finished the database schema. This allowed us to begin the task of outputting of the actual data to the system, which was by far the most challenging part of the website interface. To go about generating outputs, I assessed Simas’ code and created a general table layout. This was formed of a host of sql statements. I also needed an interactive graph solution, so I researched into using Google Graphs, I included my research on the website, and summarised the copyright statements. These were added to the site and then it was tested by myself and Simas. We spent a long time adding different hotels and checking for bugs before allowing the team to test it themselves. When we were happy with the system, I paired with Simas to write chapter 3 of the team report. We thought that all of the technical documents and explanation should be done by us, as we took on all the design, setup and implementation of the product.

Desmond: Wei: Elected as a team specialist, my role in the team is to support the team leader, project manager and the other specialists. In the early stage, we were assigned with research tasks by the team leader to fully understand the basics, idea and structure of a data scraping program. During the lecture on week 6, we were given a number of example program by our lecturer to inspire us with ideas on constructing a data scraping program. Each of us took the liberty to research on separate programs and explain how each of it works. Whenever I had any additional ideas that could help to develop our project, I would upload it into the team database on the Moodle server. Examples of the materials that I had uploaded are <http://cheapshark.com> and Razer Cortex. During one of our personal team meetings, Ben and Simas suggested on creating basic web templates based on a framework known as Bootstrap 3. As I volunteered on helping Ben out with web designs, I had made some research on the stated framework so that I would have an understanding on how it works so that it would be easier to work with it later on. On week 17 lab, we were assigned to discuss on presentations regarding IEEE Specifications among the team members. I made a summary on how to draft the specific requirements of our project and presented it to the group. This would benefit us as it would reduce the confusion of drafting the specific requirements later on. I took on the role as the product tester to make sure that our website would not have any faulty links, bugs or any safety issues. I was able to spot out a few bugs that were harmless and listed them into a document (CE101 Bug Test) that could be referred on the online team database.

# Chapter 3 Product Development (3901 words)

## 3.I An introduction to Product Development

Product development is the process of coming up with an idea for a potential product and taking it all the way to completion. This can be a new product or an upgrade/extension of an existing product, either way it will have to follow a development process for the product to be created efficiently. There are different methods for product development but they all basically break down a large task into smaller steps making the whole process more manageable. One method is known as the Waterfall method and that is what we used to develop this product. In the Waterfall method you come up with a set of requirements, then you start planning the solution, once the product is planned out you create it, and finally you test it to make sure it works. There are 2 large drawbacks to this method; if the requirements are not clear then the product may not solve the issue and secondly, any changes to the plan during implementation or testing will prove very expensive and take time.

## 3.II The Team Product

### 3.II.a The product specification

Our product is aimed at Hotel owners in general. It serves as a helpful tool to manage review sites, such as TripAdvisor. The GUI side of the program is in the form of a website, utilising HTML and PHP.

This side of the system should allow a user to log in or register and use the system. There should be a project management pane in the settings that allows a user to manage their project on the system. This allows a user to change the URL of their hotel on TripAdvisor. There should be a general settings pane that allows for other utilities such as password changing.

The project tab on the site is where all the scraped reviews will be shown. This page will allow the user to look through reviews that have been pulled from the trip advisor site, which will include the text from the reviews and the overall ratings per review. There will be features that allow users to view these in a graphical format, by using interactive graphs and charts. The user should be able to map their hotel average review rating by time in a graphical format which will allow the Hotel Owner/Manager to keep track of months that they did well in, and months they didn’t. The reviews will be sorted by time so they should be able to read up on the reviews that got them better ratings that month.

To do all of this, the scraping element will be coded using Python, and extensions such as Beautiful Soup, Requests and pymysql. These will allow for the easiest and most efficient scraping method and storage of the data, especially when it is used with the website, as were summarised by Ben after they were discussed in the CE101 Lectures [1]. Simas further went on to describe the Requests module which gave for a better understanding on how the module works. This allowed us to plan the usage of it. [2] That will be coded using HTML, PHP, JavaScript elements and SQL. Both the sides of the code will communicate with a MYSQL database.

The database will store all the data from the system. This will include some global settings, scraped reviews, projects and user data.

System will be hosted in virtual private server, so it could be easy accessed via internet connection. There will be modules and clients installed to be able to run PHP, Python and MYSQL. The user of the product will not be asked to install any specific programs and the wide range of internet browsers will be supported.

### 3.II.b The product design

The project needed a lot of time dedicated to designing the system so that both sides (Python and PHP) would work in harmony.

The project design starts with the database. To create the system we would need a database hold all of the information from TripAdvisor and for user logins. The database has to be fully functional and normalised to avoid redundant data. It would need many tables: Owner (to hold all of the information about the users); Project (to hold information about the each users project setup, such as the hotel URL); ScrapedReview (to hold all the information scraped from the physical webpages); Session (to hold information about the project status, for example, the last time it was reviewed) and finally SessionhasProject (to physically link the session and project and to keep the database normalised). A database schema was drawn up by Simas for the basic database needs before work began creating and implementing this. [4]

All of the data scraping, recording and validation will be processed on the Python side of the system. When the user enters their details and all information is validated in the GUI, a new session should be created and the Python script will be invoked. The session id, project id and the project URL should be carried over from the GUI side to the Python. At the beginning of the process, Python should scan the settings table located in the main database to adjust its configuration. The admin should be able to define how many reviews are written to the database per connection, the waiting time for each connection to avoid overloading the hotel review site servers and the method that should be used for review text extraction.   
When the required configuration is deployed, the script will check if the website is online. If the check is successful the first scan will start to determine how many pages should be scanned. If it is not the first time project has been scanned, the program will check where it finished its scan last time, so server resources are not going to be wasted and duplicates will be avoided. The main scraping process will be placed in two loops. First loop will download the whole page and will convert it in to the BeautifulSoup object. This will allow us to parse the html page and make it more manageable. The second loop should extract HTML tags were the reviews are placed. When we will have the desired data, the source will be fed to defined functions for text, rating and date extraction. All the data will be stored using tuples, which then will be added to the main list. There should be some checks implemented to control loops, as too many inactive loops could crash the system or slow it down. When the program is initiated the timestamp should be set, which will allow us to break the loop if it takes too long to complete. When all scanning is complete and all the data has been transferred to the database, the script should close all connections and update the session so the GUI part will be able to determine that the scraping is done. If the scan should fail the appropriate status code will be recorded to the session row and the appropriate message should be displayed.

Several external Python modules will have to be used. Beautiful Soup will be installed to manipulate HTML source code. This module makes this process much easier as you can convert the source code to the object and navigate between the HTML tags. Second module will be pymysql, we need to install it because Python does not have built in support as PHP does. The Requests module will be imported, so we can establish HTTP connection and download the desired website.

The GUI side of the system, which is based around a website, will need to be a fully functional website but also interact with the python code to scrape the websites. The system must allow users to login and access their project. This will entail the users going through a login process and being directed to a home page where many session variables would be started to hold details about the user. If the user does not have a login they can create one, which should update the database upon registration. Once the user has logged in they should be able to setup a project which should allow the user to select the hotel of their choice, and also edit basic settings, such as passwords and also the hotel URL address, the system should constantly update the database with new settings or to check entered passwords and other elements.

When the user has logged in, they should be presented with different pages, in an attractive layout using tabs, and should be able to switch between pages, giving an introduction to the system, information about the licenses that we have had to use to create the system and information about the project and maybe some links to documents such as the finalised team report and design documents.

When the user selects the scrape tab, they should be given the choice to use old data, or refresh the database. This is when the PHP should interact with the python script, and begin to scrape the hotel review site for information. The user should be informed that the system is working, because it will not be instantaneous due to the delay in the queries to the review site. Once the python script has finished, the database will be up to date and the system should query this to get data on the users reviews, ratings and statistics from the review site and present this all in one neatly presented page. There should be graphical elements such as graphs and tables and also the plain text reviews all laid out in a tabular fashion. The user should be able to filter reviews by date and easily analyse their reviews by using the predefined elements on the page.

Once the user has finished looking at their reviews, they should be able to simply logout of the system by clicking on the logout button. This should destroy the session data on the system.

To do this we have chosen to use the Bootstrap 3 framework available on an open source license. This will allow for maximum compatibility across many devices and browsers, new or old. It will also allow the system to look modern and comes with many features such as pre-defined div elements, Glyphicons and JavaScript elements such as modals and interactive graphs and charts. Using this framework will allow for the system to look professional and modern with less time dedicated to the styling. This was summarised by Ben and given as a presentation to the group, he mentioned some of the features that would be utilised and gave some examples. [5]

The language of choice for the website will be PHP. This is a great language for creating dynamic websites as it is very versatile, secure and also is widely available across many webservers that we will have access to. There will be elements created using JavaScript and a html error page and index.

To make an efficient GUI, Jeffery took some time to produce a quick document that gave us an idea as to the layout for our GUI. To do this he took time to look at commonly used, popular review sites for Hotels, which were TripAdvisor and Booking.com and summarised what makes them a good and easy to use system. [6] To give us an idea of other Hotel Review sites and how they looked, Emmanuel provided a collection of screenshots from live sites [7] and Ted also provided layouts for other online sites that scraped data together, which helped with the design for the pages on the system that provided user output. [8][9]

The database system we are using will be PHP MYSQL. This will allow the utilities of MYSQL to be accessible on a webserver. The server that we host on will define the specific software that is used to manage the database, but we know that it will be local to the system, and will work with both PHP and Python.

A web server with a specific configuration will be required. The project will be hosted in virtual private server running an Ubuntu (Linux branch) operating system. This operating system has been chosen because it's free of charge, easy to configure and properly maintained. In this machine we will install and run NGINX (an alternative to Apache) as a HTTP server, so our project will be able to deliver the pages to the client and a MYSQL server to be able to store the reviews and customer data. It is important to pay good attention to the MYSQL configuration as it will be used heavily and there would be moderate amount of data stored inside. As well, we will have to install Python with all required modules. There are few Python module managers available, so it will make the process easier. To enable GUI we will have to install and configure PHP-fpm server. There would be additional configuration required to NGINX server, so it can use PHP-fpm server and deliver the PHP pages. In addition to that minimal server security will have to be ensured – we will have to enable and configure firewall, set up appropriate access levels to users and make sure that the operating system with all installed clients are updated regularly.

### 3.II.c The product implementation

Implementation of the product involved the purchase and setup of a small web server. This allowed us to host the product online, where it would be stored for general use. This is a better solution than having a local based program as we would not need to install any modules to run the system.

The code itself was created in a procedural fashion, because both of the programmers were more familiar with this layout for code. The PHP could have been coded in a procedural fashion, however this would have only worked best for the MYSQL queries, however they were laid out into pre-configured variables anyway. The python code worked best in a procedural layout, as it was easiest just to pass lists around, and not objects, due to the formatting each time a query would need to be executed.

There were many functions created for the Python side of the system, more than one for each job, as it was easier to share functions between an elements of the code. These functions included review\_text, long\_text, mysql\_session\_update and connect. These are very self-explanatory, but are explained in Simas’ list of functions document and the code. [10][11] For the PHP side of the system, there were a few functions created, such as, checkadmin, checkurl, checksession, pwcheck, pwhash and checklogin. These all were given logical names and their purpose commented in on the code. [12] The main body of ‘functions’ for the PHP was a large if loop. It utilised the form elements on the website, and checked for the form that was submitted against its given name in the POST data. This allowed for specific parts of the code to be executed.

Simas took control of the server setup, being very experienced with Linux, this was an apt job for him. A server from BalticVps, a very reliable provider located in Lithuania was chosen to host. For the beginning the basic type of Virtual Private Server (VPS) looked completely enough. We installed an open source operating system Ubuntu 14.04 64 bit edition. The system was upgraded and all required modules were installed via apt-get application manager.

Firstly Apache2 was installed so we can deliver the web pages. We used a default configuration, as there were no additional modifications needed. After that the PHP5 client and MySQL server was installed with all required modules. We had to have those modules in our server to be able to run GUI. The Apache2 has its own module managing service, so this helped us out a lot. When we had all the parts which were required for GUI, the Python3.4 was installed so we would be able to run the scrapping script. To save some time, the PIP was installed as well. This small application lets us to add most popular Python modules much easier. With help of PIP all required modules were installed. There were some changes made to the default firewall so the scraper can be run smoothly.

The server was documented by Simas, and a small document was created telling our team how to get onto the server and view the files. [13]

Once the server had been setup, to use the system, the end client has to just point their browser URL to the IP or registered domain. The one we registered was a .tk domain, [www.hotelscraper.tk](http://www.hotelscraper.tk) [14] [15].

Everything worked well with the system except for the database connections. This was the only code we had to change when moving from local machine hosts to an online one. We had to change the login details for the MYSQL database to match the one we had setup on our server. You can see this change at the top of the PHP constants page, where it lays out the connection details in a small variable. [16]

### 3.II.d The product testing

The system needed constant testing whilst being created. This fell under general debugging. Python allows the use of IDLE, which will dynamically check the code whist it is being interpreted. This allowed for great help when creating the Python side of the system. However it was not such help when using the Python modules that were imported and a lot of time was spent analysing and learning how to use the specific modules themselves.

There is no syntax checker that comes stock with PHP, so that was created without one, but being experienced with PHP helped. There were many bugs on the way with the website side of the system, but all were ironed out by debugging and testing as the development progressed. To be able to run the system, a local host server had to be setup, that was capable of processing PHP scripts and also had a PHP MYSQL database setup. phpMyAdmin was used for the database setup on the website side of the system whilst the development was ongoing. This was a helpful tool, as SQL queries could be tested and checked using the software.

Whilst coding the system, we used a form of agile development which entailed test-driven development within each cycle. This allowed us to create the system in smaller chunks. The code was split into 2 halves, the PHP and the Python. It allowed the programming specialists to focus one part separately. When developing, the code was updated in small cycles, and at each stage, the code was tested for bugs. It allowed for a more bug free product to be rolled out at the end of development, and helped when we came to join the systems together and display all of the scraped data we had formulated. For example, the cycle where the Settings pages on the website were developed, was done in one chunk, and that was split down between each element on the page. So the code would be created, tested until the developer thought it was bug free and uploaded. These were all tested at the end by another team specialist. So the individual development of functions was test driven, however the segmentation of sections was agile development.

Once the system was live, Desmond took the job of testing the system on multiple browsers. The first main bug that he found, was that the system would not work on Internet Explorer. This is a big bug, but I believe with some configuration it would work. [17] When developing it would work on the local machine host, so it may be a server issue. However with other browsers it worked well. All fields were tested, Desmond tried to enter incorrect values in fields. He documented all of the errors that he found. During his testing, he found a few minor errors, however most of the system worked the way that it should. The first error he documented [18] was that when a non TripAdvisor URL was inserted in the settings page, the URL was not accepted, but an error was not displayed [19] shows a bug that when the user tries to change their password, if the new password does not match the confirmation field, the password is not changed, but no error message is shown. These were the only bugs found in the system. There may be a few, but most are very minor and do not affect the system at all, except from displaying error messages. The big bug was that the system would not work with Internet Explorer, which was bad news because the whole idea of using the Bootstrap framework was that the html would work on any browser.

## 3.III Context

### 3.III.a Legal matters

When we think of legal matters, instantly the principle of ‘intellectual property’ is introduced. Here, we have many rules to abide by in order to successfully develop a product, one of which is **copyright**, this is the authenticity of the actual idea behind a product. In our case, as far as we were aware we had thought of an entirely unique program which is not based on any prior creation so we were not breaching any terms or conditions. Furthermore there was the issue of **patentability**, this is where the questions ‘is the invention/creation new?’ and ‘what sets it apart from the rest?’ come into play, the simple answer was yes because as opposed to creating merely another single hotel review site, we were using multiple to get broader reviews and comments.

To aid us develop our product we used further programs such as Google charts and bootstrap, Google Charts helped us with visualizing data on our website, it provided several ready-made chart types.[20] In order to use this we were given the responsibility of protecting the privacy of users (Those who submitted the hotel reviews) as our data manipulation was based on their feedback. Another program we used to help with the aspect of HTML was BootStrap, in order to use this however we needed to provide a purple and white trademark logo. [21]

### 3.III.b Ethical matters

Ethics is defined as a ‘branch of philosophy that defines what is good for the individual and for society’, applying this to our project the only possible setback we may have had in terms of ethics was the lack of consent given by the actual reviewer, as we were using many reviews from different sites. Asides from that, there were no obvious ethical issues in our program which would have severely affect the production.

### 3.III.c Health & safety matters

Health and safety is explained as **ALARP** (As Low As Reasonably Practicable) and SFAIRP (So Far As Is Reasonably Practicable). ALARP is a form of reducing risk but in our case, our project was based around a hotel-review sites so there was no direct danger imposed on any user of the program. **SFAIRP** is what we actually mean by ‘reasonably practicable’, it involves a healthy balance between the ‘quantum of risk’ vs ‘the sacrifice involved in averting risk’, if there is gross disproportion then the project as a whole is under threat but as expressed before, there was no real hazard caused by our project so the effort put into reducing hazard was minimal and this mirrored the size of the potential actual risk/hazard/danger.

References:

1. CE101 lecture – Week 6 lecture from CE101 course – Anthony Vickers

(<https://moodle.essex.ac.uk/mod/book/view.php?id=208150&chapterid=1893>)

1. Extension summaries from w6 lecture.docx – Description of python extensions from CE101 Lecture – Ben Higgs

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/88347/extension%20summaries%20from%20w6%20lecture.docx>)

1. requests\_precis.pdf – Precis of how to use Python Requests module – Simas Kairys

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/89390/requests_precis.pdf>)

1. Mysql.png – Image of first proposed Database Schema – Simas Kairys

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/96791/mysql.png>)

1. Bootstrap pres ben.pptx - Bootstrap3 summary – Ben Higgs

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/90992/bootstrap%20pres%20ben%20.pptx>)

1. Review website research – Hotel Review site analysis – Jeffrey Chujor

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/89928/Review%20website%20research.docx>)

1. Website Design.docx – suggestions for web design - Emmanuel Odame-Asante

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/88295/Website%20design%20.docx>)

1. Allkeyshop URL and screenshot – Ted Fey

(<http://www.allkeyshop.com/blog/> , <https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/88065/allkeyshop.jpg>)

1. GoogleShopping URL and screenshot – Ted Fey

(<https://www.google.co.uk/shopping> , <https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/88069/googleshopping.jpg>)

1. Python code final.zip – Zip of finished python code – Simas Kairys

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/97954/PYTHON%20code%20final.zip>)

1. List.odt - List of functions in the Python code – Simas Kairys

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/97942/list.odt>)

1. PHP code final.zip – zip of finished PHP code- specific referenced documents inside (functions.php and constants.php) – Ben Higgs

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/97950/PHP%20code%20final.zip>)

1. Accessing\_server\_pdf.pdf - Document explaining how to access the web server – Simas Kairys

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/95727/accessing_server_pdf.pdf>)

1. Hotelscraper domain link – Ben Higgs - <http://www.hotelscraper.tk/>
2. IP link to the server (no DNS) – Ben Higgs - <http://185.69.53.182/login.php>
3. PHP code final.zip – zip of finished PHP code- specific referenced document inside (constants.php) – Ben Higgs

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/97950/PHP%20code%20final.zip>)

1. CE101 bug test.docx - CE101 bug test page 16 – Desmond Wei Chua

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/96707/CE101%20Bug%20Test.docx>)

1. CE101 bug test.docx - CE101 bug test page 9 part 4)a – Desmond Wei Chua

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/96707/CE101%20Bug%20Test.docx>)

1. CE101 bug test.docx - CE101 bug test page 13 part 4)e – Desmond Wei Chua

(<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/96707/CE101%20Bug%20Test.docx>)

1. Google charts (<https://developers.google.com/chart/interactive/docs/security_privacy>)
2. Bootstrap (<http://getbootstrap.com/about/> )

# Chapter 4: Project Management (962 words)

## 4.I An introduction to Project Management

Whenever a project of any kind is being developed it has to be managed in order for everything to run smoothly. Effective project management will allow you to predict costs, time required, realistic deadlines, backup plans if things go wrong, etc. Most projects of a reasonable size will have a designated project manager who is in charge of planning out the product’s development cycle and keeping track of everything happening in order to make sure there is a smooth workflow and identify problems as soon as possible. A project will get split into a number of phases or processes that go from the start to the end, an example could be: [1]

1. Identify audience and/or requirements – Is it for a specific demographic? Does it rely on other equipment?
2. Plan the solution – An idea that will solve the problem at hand.
3. Set costs, deadlines and assess risks – Budgets; when does it need to be finished? What could slow us down?
4. Put the team together, each with a specific role – e.g. Leader, programming specialist, etc.
5. Supervise the project while it is created – make sure all targets are met and identify problems early.
6. Test the product and release it – Does it work as intended? Is it secure and safe?

The project manager’s job throughput these processes is to plan budgets and deadlines, log anything that gets done and who it was done by and also adjust the plans if anything changes so that the product can continue with efficient development. Risk management is also a part of a project manager’s job as risks can affect the development process or may have a domino effect on other parts of the project. Through detailed planning, problem reporting and logging of information the risks can be minimised and if something does go wrong it’s much easier to adjust and recover.

All of this planning and logging of work being done may seem like additional work or even unnecessary sometimes, but the further you get into a product’s development, the more useful it starts to become, this is often referred to as the “pain curve” [1]. There are many tools available to project managers to keep track of everything, one of the oldest and most versatile methods is keeping a logbook of everything that is happening with everything dated and this si a tool we used greatly with Microsoft’s OneNote program [2]. Another great tool is called a Gantt chart which keeps track of how many hours each team member is doing, what tasks they have been assigned, the cost of each resource and provides a graphical timeline of the product development process (ours is located in appendix B at the end of this report).

References:

1. Dr Anthony Vickers (2014). Week 8 – Lecture: Project management. Available: <https://moodle.essex.ac.uk/mod/book/view.php?id=208150&chapterid=1916>
2. Mr Ted Fey (2015). CE101 Logbook – Team H. Available: <https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/97990/CE101%20Logbook%20-%20Team%20H.pdf>

## 4.II Project Management Report

### 4.II.a A description of the Gantt chart[[1]](#footnote-1)

The Gantt chart is a break-down of what each team member has been working on between each meeting. I have broken it down into 4 main parts which each represent a phase in the product’s development: Planning, Design, Implementation and Completion. We did jump back and forwards a little between design and implementation as Simas and Ben had already started on some basic Python code before we had quite finished the overall design of the product. Each of the tasks on the Gantt chart go by a meeting-to-meeting basis, so many of them would last 2 weeks as that is the gap between our main lab meetings, however the 1 week tasks are a result of other meetings that we arranged in our own time. One of the mistakes we made early on in the project was moving on too quickly and not completing previous work we had started, an example of this on the Gantt chart is the IEEE précis; as you can see we had to go back at the end of the design phase to finish it off so we could finally write our product specifications. There was a lot of important tasks left in the final weeks of the project but by allocating the tasks equally according to each team member’s strengths we were able to wrap it all up in time to hit our deadlines for the product presentation and submission of the final team report.

### 4.II.b An evaluation of the project management

The management of our project started really strong in the first couple of weeks, the logbook was organised and we met our first deadlines without problems. Admittedly in the next few weeks we didn’t achieve much, deadlines were not being met, communication was minimal and we were not taking advantage of the team forum and the other Moodle services. Once we realised we were falling behind I think it was a bit of a wakeup call and we organised our own meetings, started uploading files to Moodle and setting SMART (specific, measurable, achievable, relevant and time-related) objectives to each team member, all whilst keeping in touch using WhatsApp, the team forum and having meetings. The more organised and better managed we were, the more work we got done. Since we started writing the team report and finishing up our product our project management has really improved, everyone knows exactly what they are doing and when it must be completed by. As project manager I think the weakest part of my performance is my lack of consistency, having missed a few meetings and there was a period where I didn’t keep up with my duties with the speed and quality that I should have. To improve the project management of the team I think communicating more and working with everyone to set clear goals every week would have made the project much easier to complete.

# Chapter 5: Conclusions (532 words)

## Team H has developed an excellent product; of five months of researching, planning, developing and testing we have a fully polished product. The Hotel Scraper. Our product scored second best in the product demonstration only losing by one point to the first term. Our Hotel Scraper accurately imports data from websites such as trip advisor and successfully analyses the data and exports on a user friendly style. On our easy to use website we have various graphs visualizing the reviews of a specific hotel. Also implanted is the last few comments on the specific hotel. Not did our product grow well but as a team of six we’ve worked well with each other. Even though at the start of the year our communication and team working skills wasn’t at its optimum, we have definitely improved on those traits as the months have passed. Luckily for us we decided on the hotel scraper idea very early which gave as a huge advantage on actually researching and looking at developing the product. Without the dedication and hard work of the team there will not be a product like this. There are many stories of how teams fail miserably when there is not enough communication and team spirit. Thankfully our team was not like that. If we had to go back and change anything about our development process of Team H as a team it is agreed that at the early stages we should of communicated more and started important pieces of work such as the Team Report. However in general we do not think there are many flaws in Team H. Every single member in this team has contributed in this project. Obviously there are specific members who have spent more hours on the coding side of the project than others but that is mainly due to the fact that the coders really knew what they were doing and wanted to complete the section themselves. A huge advantage in this group is that two of our team specialist had grand experience in coding. They have completed projects before. So that really increased the quality of our product in this project. As a team we all worked hard to complete our section to our best however, there were some performance dips especially when we had work from other modules that we needed to complete. In general the performance was always above average. It is agreed that our strongest advantages in the team was our work ethic and communication. The Team was always working on the project part time so it can be as best as it can be. Originally we did want extra features on the Scraper like Twitter and Facebook incorporation but due to time issues and other deadlines that needed to be completed this idea had to be scraped. This is not a huge disadvantage we do not have this feature in our program, it was just extra flair that could make the program extra exotic. However to conclude the Team report Team H was worked very hard on both the program and Report to deliver the best project to client Mr Vickers. The team and product as progressed fantastically from October.

# Appendix

## A. Database normalization

Normalisation was applied on the database of the data scrapper to reduce redundancy caused by repetition of data. There are 4 normal forms used in the database. The First Normal Form (1NF) has a condition that must be fulfilled, which is the values in each cell of the tables must be atomic. There would be no repeating groups. A relationship is in Second Normal Form (2NF) if it is in 1NF and every non-prime attribute of the relationship is fully functional dependant on each candidate key of the relationship. The relation SesionHasProject has 2 Primary Keys (session\_id and project\_id). By splitting it into two relations, we will form two new relations which are in 2NF. A relationship is in Third Normal Form (3NF) if it is in 2NF and every non-prime attribute of R is non-transitively dependent on each candidate key of the relationship. For example, the relation Sesion can be derived the following transitive dependency. [id 🡪 trip\_adv\_url, booking\_url 🡪 keyword]. By splitting it into two relations, we can convert the relations into 3NF; [id, keyword] and [keyword, trip\_adv\_url, booking\_url].

**TABLES :**

**Owner**

1. **Groups :** [ownerId : INT], [email : VARCHAR(80)], [passwordHash : CHAR(255)]
2. **Superkeys :** (ownerId), (email), (passwordHash), (ownderID, email), (ownerId, passwordHash), (email, passwordHash)
3. **Primary key :** ownderId

**Project**

1. **Groups :** [projectId : INT], [website : VARCHAR(255)], [owner\_project : INT]
2. **Superkeys :** (projectId), (website), (owner\_project), (projectId, website), (projectId, ownder\_project), (website, owner\_project)
3. **Primary key :** projectId
4. **Foreign key** : owner\_project 🡪 Owner(ownerId)

**Sesion**

1. **Groups :** [id : INT], [status : VARCHAR(1)], [start\_time : TIMESTAMP], [trip\_adv\_url : VARCHAR(255)], [booking\_url : VARCHAR(255)], [keyword : VARCHAR(50)], [projectId : INT]
2. **Superkeys :** (id), (status), (start\_time), (trip\_adv\_url), (booking\_url), (keyword), (projectId), (id, status), (id, start\_time), (id, trip\_adv\_url), (id, booking\_url), (id, keyword), (id, projectId), (status, start\_time), (status, trip\_adv\_url), (status, booking\_url), (status, keyword), (status, projectId), (trip\_adv\_url, booking\_url), (trip\_adv\_url, keyword), (trip\_adv\_url, projectId), (booking\_url, keyword), (booking\_url, projectId), (keyword, projectId)
3. **Primary key :** id
4. **Foreign key :** projectId 🡪 Project(projectId)

**ScrapedReview**

1. **Groups :** [review\_id : INT], [sesion\_id : INT], [sourc : VARCHAR(60)], [review\_text : LONGTEXT], [rating : INT], [project\_id : INT], [original\_url : VARCHAR(255)], [date\_published : DATE], [title : VARCHAR(60)], [scraped : TIMESTAMP], [originalid : INT]
2. **Superkeys :** (review\_id), (sesion\_id), (sourc), (review\_text), (rating), (project\_id), (original\_url), (date\_published), (title), (scraped), (originalid), (review\_id, sesion\_id), (review\_id, sourc), (review\_id, review\_text), (review\_id, rating), (review\_id, project\_id), (review\_id, original\_url), (review\_id, date\_published), (review\_id, title), (review\_id, scraped), (review\_id, originalid), (sesion\_id, sourc), (sesion\_id, review\_text), (sesion\_id, rating), (sesion\_id, project\_id), (sesion\_id, original\_url), (sesion\_id, date\_published), (sesion\_id, title), (sesion\_id, scraped), (sesion\_id, originalid), (sourc, review\_text), (sourc, rating), (sourc, project\_id), (sourc, original\_url), (sourc, date\_published), (sourc, title), (sourc, scraped), (sourc, originalid), (review\_text, rating), (review\_text, project\_id), (review\_text, original\_url), (review\_text, date\_published), (review\_text, title), (review\_text, scraped), (review\_text, originalid), (rating, project\_id), (rating, original\_url), (rating, date\_published), (rating, title), (rating, scraped), (rating, originalid)
3. **Primary key :** review\_id
4. **Foreign key :** sesion\_id 🡪 Sesion(id), project\_id 🡪 Project(projectId)

**SesionHasProject**

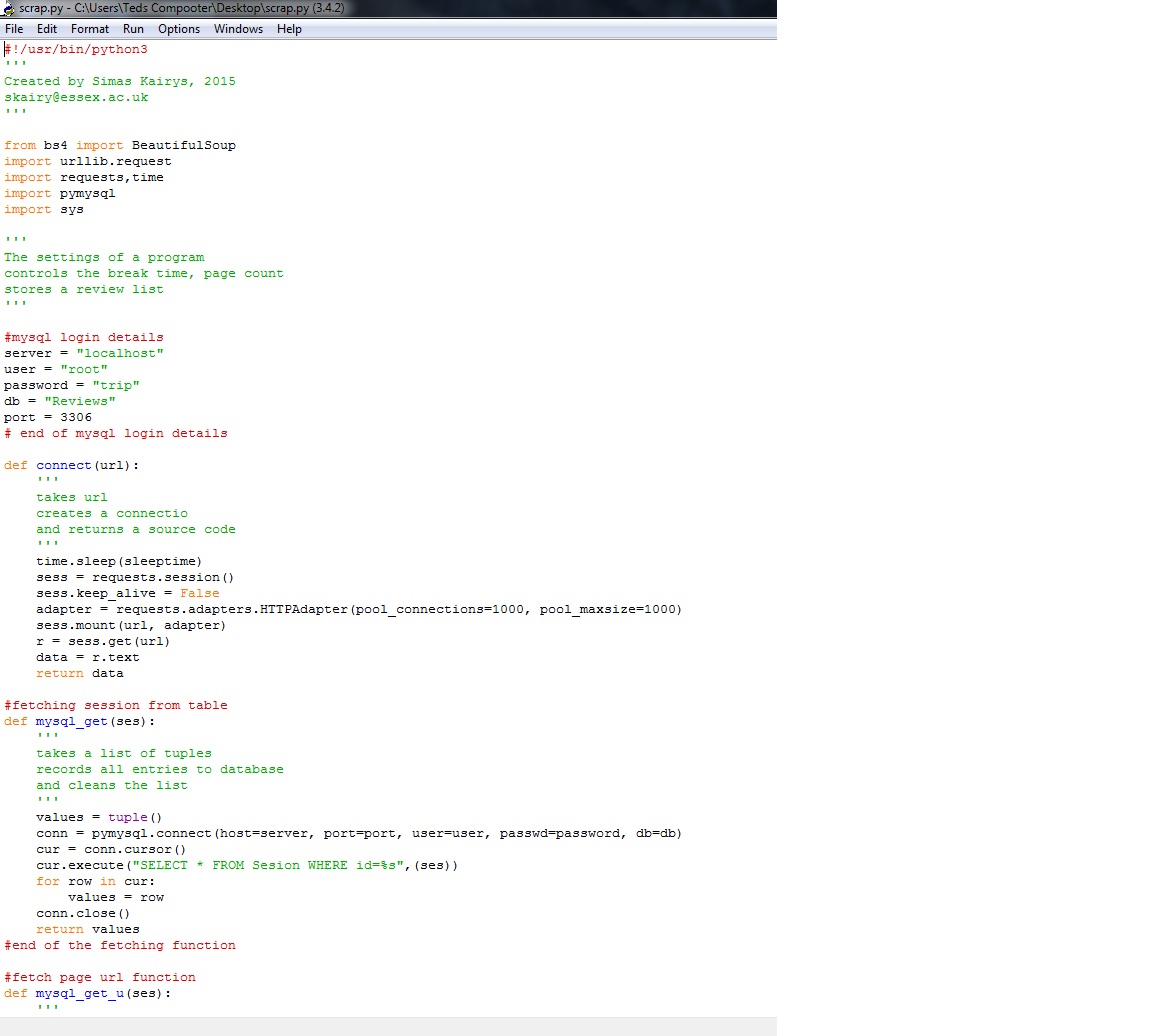
1. **Groups :** [sesion\_id : INT], [project\_id : INT]
2. **Superkeys :** (sesion\_id), (project\_id), (sesion\_id, project\_id)
3. **Primary keys :** sesion\_id, project\_id
4. **Foreign keys :** sesion\_id 🡪 Sesion(id), project\_id 🡪 Project(projectId)

**GlobalSettings**

1. **Groups :** [setup\_id : INT], [admin\_id : INT], [loop\_timeout : INT], [sleep\_time : INT], [database\_write\_quota : INT]
2. **Superkeys :** (setup\_id), (admin\_id), (loop\_timeout), (sleep\_time), (database\_write\_quota), (setup\_id, admin\_id), (setup\_id, loop\_timeout), (setup\_id, sleep\_time), (setup\_id, database\_write\_quota), (admin\_id, loop\_timeout), (admin\_id, sleep\_time), (admin\_id, database\_write\_quota), (loop\_timeout, sleep\_time), (loop\_timeout, database\_write\_quota), (sleep\_time, database\_write\_quota)
3. **Primary keys :** setup\_id

**Foreign keys :** admin\_id 🡪 Owner(ownerId)

## B. Python Code



## py 2

## py 3

## py 4

## py 6py 5

## B. Team effort summary table

|  |  |  |
| --- | --- | --- |
|  |  |  |
|
| py 7 |
|  |

## 

## C. Project management Gantt chart







## D. Team effort summary table

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Meeting Attendance** | **Number of references added to the database** | **Number of précis added to the database** | **PowerPoint presentation given to team in week 19** | **Agendas in team logbook** | **Minutes in team logbook** | **Discussions added to the Team forum[[2]](#footnote-2)** | **Product Development** | **Report Writing** |  |
| **Team Member** | **Role** | 0-10 | Rank order[[3]](#footnote-3) | Rank order | 0 or 10 | 0-10 | 0-10 | Rank order | Rank order[[4]](#footnote-4) | Rank order3 | TOTAL |
| Emmanuel | **Leader** | 10 | 6 | 6 | 10 |  |  | 4 | 2 | 6 | 44 |
| Ted | **Project Manager** | 5 | 6 | 6 | 10 | 10 | 9 | 5 | 1 | 5 | 57 |
| Simas | **Specialist** | 9 | 5 | 6 | 10 |  |  | 6 | 6 | 5 | 47 |
| Benjamin | **Specialist** | 10 | 6 | 5 | 10 |  |  | 5 | 6 | 6 | 48 |
| Jeffrey | **Specialist** | 6 | 6 | 5 | 10 |  |  | 3 | 1 | 4 | 38 |
| Wei | **Specialist** | 10 | 6 | 5 | 10 |  |  | 2 | 3 | 5 | 41 |

## E.PHP code

Underneath will be a selection of the PHP code.

Included will be Functions.php (where all the main processing goes on for the script), constants.php (to show more functions and system constants), login.php (to show the login procedure), settings.php (to show all the settings that you can change as a user) and scrape.php (to show the procedure for outputting and formatting the scraped data).

The rest of the code can be seen in PHP code final.zip, (<https://moodle.essex.ac.uk/pluginfile.php/323269/mod_data/content/97950/PHP%20code%20final.zip>)

########################## FUNCTIONS.PHP ##############################

<?php

include'constants.php';

if (session\_status() == PHP\_SESSION\_NONE) {

session\_start();

}

//used to check if session has been setup by user before hand

function checksession($userid,$con){

$site = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT website FROM Project WHERE owner\_project='$userid'")) ['website'];

if($site!=null){

$\_SESSION["ScrapeURL"] = $site;

header('Location:home.php');

}

else{

header('Location:createsession.php');

}

}

//used to check the url that the user inputs, this checks if it is a valid hotelreview

function checkurl($url){

if (strpos($url,'http://www.tripadvisor.co.uk/Hotel\_Review-') !== false) {

return true;

} else {

return false;

}

}

//checks if admin is logged in, if not, page goes back to login

function checkadmin($userid,$con){

$admin = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT admin\_id FROM GlobalSettings WHERE setup\_id=1")) ['admin\_id'];

if ($userid == $admin){return true;}

else{return false;}

}

//checks users login and verifys the PW hash

if (isset($\_POST['login'])) {

if($\_POST['email']!='' and $\_POST['password']!=''){

$email=$\_POST["email"]; $password=$\_POST["password"];

if (pwcheck($password,mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT passwordHash FROM Owner WHERE email='$email'")) ['passwordHash'])==1){

$\_SESSION["error"]="none";

$que=mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT ownerId FROM Owner WHERE email='$email'"));

$userid=$que['ownerId'];

$\_SESSION["userid"]=$userid;

$\_SESSION["email"]=$email;

if (checkadmin($userid,$con)){

$\_SESSION["admin"]=true;

header('Location:home.php');

}

else{

$\_SESSION["admin"]=false;

checksession($userid,$con);

}

}

else{

$\_SESSION["error"]="nomatch";

header("Location:login.php");

}

}

else{$\_SESSION["error"]="fieldcheck";header("Location:login.php");}

}

//redirects users to the registry page

if (isset($\_POST['registryform'])) {header("Location:registry.php");}

//registers the users on the DB, does all the appropriate checks.

if (isset($\_POST['register'])) {

if($\_POST['email']!='' and $\_POST['password']!=''){

$email=$\_POST["email"]; $password=$\_POST["password"];

$query=mysqli\_num\_rows(mysqli\_query($con,"SELECT email FROM Owner WHERE email='$email'"));

if ($query==0){$password=pwhash($password); mysqli\_query($con,"INSERT INTO Owner (email,passwordHash) VALUES ('$email','$password' )");

$\_SESSION["userid"]=mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT ownerId FROM Owner WHERE email=$email"))['ownerId'];

$\_SESSION["error"]="new";

header('Location:login.php');

}

else if($query!=0){$\_SESSION["error"]="emailexist"; header('Location:registry.php');}

}

else{$\_SESSION["error"]="fieldcheck";header('Location:registry.php');}

}

//changes password

if (isset($\_POST['changepw'])) {

if($\_POST['password1']!='' and $\_POST['password2']!='' and $\_POST['password3']!=''){

$password1=$\_POST["password1"]; $password2=$\_POST["password2"]; $password3=$\_POST["password3"]; $email=$\_SESSION["email"];

if ($password2 != $password3){$\_SESSION["error"]="newpwmatch"; header("Location:settings.php"); }

else if (pwcheck($password1,mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT passwordHash FROM Owner WHERE email='$email'")) ['passwordHash'])!=1){

$\_SESSION["error"]="oldpwmatch";

header('Location:settings.php');

}

else{$hash=pwhash($password2);

mysqli\_query($con,"UPDATE Owner SET passwordHash='$hash' WHERE email='$email'");

$\_SESSION["error"]="pwupdated";

header("Location:settings.php");

}

}

else{$\_SESSION["error"]="pwsnotthere";header("Location:settings.php");}

}

//adds the url to the DB in the correct location

if (isset($\_POST['addurl'])) {

if(checkadmin($\_SESSION["userid"],$con)==true){

header('Location:home.php');

}

else{

$URL=$\_POST["URL"];

if ($URL!=''){

if (checkurl($URL)==true){

$id=$\_SESSION["userid"];

mysqli\_query($con,"INSERT INTO Project (website,owner\_project) VALUES ('$URL','$id')");

checksession($id,$con);

}

else{$\_SESSION["error"]="invalidurlformat";

header('Location:createsession.php');}

}

else{

$\_SESSION["error"]="invalidurl";

header('Location:createsession.php');

}

}

}

//changes the url, used when user changes their url

if (isset($\_POST['changeurl'])) {

$URL=$\_POST["newurl"];

$id=$\_SESSION["userid"];

if ($URL == mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT website FROM Project WHERE owner\_project='$id'")) ['website']){

$\_SESSION["error"]="sameurl";

header('Location:settings.php');

}

else if (checkurl($URL) == false){

$\_SESSION["error"]="invalidurl1";

header('Location:settings.php');

}

else if ($URL!=''){

mysqli\_query($con,"UPDATE Project SET website='$URL' WHERE owner\_project='$id'");

$\_SESSION["error"]="updatedurl";

$\_SESSION["ScrapeURL"] = $URL;

header('Location:settings.php');

}

}

//changes the url of a specific user, this is the admin version of changeurl

if (isset($\_POST['changeuserurl'])) {

$URL=$\_POST["newurl1"];

$id=$\_POST["emailchoice"];

$id=mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT ownerId FROM Owner WHERE email='$id'")) ['ownerId'];

if ($id=='na'){

$\_SESSION["error"]="emailselect";

header('Location:settings.php');

}

else if ($URL == mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT website FROM Project WHERE owner\_project='$id'")) ['website']){

$\_SESSION["error"]="sameurl1";

header('Location:settings.php');

}

else if (checkurl($URL) == false){

$\_SESSION["error"]="invalidurl1";

header('Location:settings.php');

}

else if($URL!=''){

mysqli\_query($con,"UPDATE Project SET website='$URL' WHERE owner\_project='$id'");

$\_SESSION["error"]="updatedurl1";

header('Location:settings.php');

}

}

//this is used then the admin changes system settings, it finds what form inputs are used and sets them in the db and success messages.

if (isset($\_POST['changeadminsettings'])) {

if (isset($\_POST['reset'])){

mysqli\_query($con,"UPDATE GlobalSettings SET loop\_timeout=600,sleep\_time=5,database\_write\_quota=5 ");

$\_SESSION["error"]="resetDBsettings";

header('Location:settings.php');

}

else{

if($\_POST['looptimeout']!=''){

$q = $\_POST['looptimeout'];

mysqli\_query($con,"UPDATE GlobalSettings SET loop\_timeout=$q ");

$\_SESSION["error"]="looptime";

}

if($\_POST['sleeptime']!=''){

$q = $\_POST['sleeptime'];

mysqli\_query($con,"UPDATE GlobalSettings SET sleep\_time=$q ");

if($\_SESSION["error"]=="looptime"){

$\_SESSION["error"].="sleeptime";

}

else{

$\_SESSION["error"]="sleeptime";

}

}

if($\_POST['writequota']!=''){

$q = $\_POST['writequota'];

mysqli\_query($con,"UPDATE GlobalSettings SET database\_write\_quota=$q");

if(strpos($\_SESSION["error"],"looptime") !== false || strpos($\_SESSION["error"],"sleeptime") !== false){

$\_SESSION["error"].="writequota";

}

else{

$\_SESSION["error"]="writequota";

}

}

header('Location:settings.php');

}

}

//this is used when the user chooses to view data already stored in the db about their hotel.

if (isset($\_POST['old'])){

$userid = $\_SESSION['userid'];

$s = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT sesion\_id FROM SesionHasProject JOIN Project ON SesionHasProject.project\_id = Project.projectId WHERE owner\_project='$userid'")) ['sesion\_id'];

$\_SESSION['scanview']=true;

$\_SESSION['sesion']=$s;

header('Location:scrape.php');

}

//this is used when the admin chooses to view data already stored in the db about a user hotel.

if (isset($\_POST['showuserhotel'])){

$email = $\_POST['emailchoice'];

$userid = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT ownerId FROM Owner WHERE email='$email'")) ['ownerId'];

$\_SESSION['usersid'] = $userid;

$s = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT sesion\_id FROM SesionHasProject JOIN Project ON SesionHasProject.project\_id = Project.projectId WHERE owner\_project='$userid'")) ['sesion\_id'];

$\_SESSION['scanview']=true;

$\_SESSION['sesion']=$s;

header('Location:scrape.php');

}

//used when the user refreshes data in the db about their hotel

if (isset($\_POST['new'])){

$id = $\_SESSION["userid"];

$pid = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT projectId FROM Project WHERE owner\_project='$id'")) ['projectId'];

mysqli\_query($con,"UPDATE Project SET last\_scan=null WHERE projectId='$pid'");

$lasttime = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT \* FROM Sesion WHERE projectId='$pid' ORDER BY start\_time DESC LIMIT 1"));

$checklaststatus = $lasttime ['status'];

//checks if user has unfinished scans in db, if they do, loading page is loaded, this stops system overload, else, new scan started

if($checklaststatus == 'p' || $checklaststatus == 's'){

$\_SESSION['sesion']=$lasttime ['id'];

$\_SESSION['loading'] = true;

$\_SESSION['startscrape']=true;

header('Location:loading.php');

}

else{

mysqli\_query($con,"INSERT INTO Sesion (start\_time,projectId) VALUES (NOW(),$pid)");

$s = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT id FROM Sesion ORDER BY start\_time DESC LIMIT 1")) ['id'];

mysqli\_query($con,"DELETE FROM SesionHasProject WHERE project\_id='$pid'");

mysqli\_query($con,"INSERT INTO SesionHasProject VALUES ($s,$pid)");

shell\_exec("/var/www/html/py/scrap.py ".$s ." > /dev/null 2>/dev/null &");

$\_SESSION['sesion']=$s;

$\_SESSION['loading'] = true;

$\_SESSION['startscrape']=true;

header('Location:loading.php');

}

}

//this is used when the admin deletes the user

if (isset($\_POST['removeuser'])){

$email = $\_POST['emailchoice'];

$id= mysqli\_fetch\_assoc(mysqli\_query($con, "SELECT ownerId FROM Owner WHERE email='$email'"))['ownerId'];

mysqli\_query($con,"DELETE FROM Project WHERE owner\_project='$id'");

mysqli\_query($con,"DELETE FROM Owner WHERE email='$email'");

$\_SESSION["error"]="removeduser";

header('Location:settings.php');

}

?>

######################## CONSTANTS.PHP ##########################

<?php

//max execution time set for php page before error

ini\_set('max\_execution\_time', 300);

//default database connection settings are stored here, this makes it easier to start connection.

$con=mysqli\_connect("localhost","root","trip","Reviews");

if (mysqli\_connect\_errno($con))

{

echo "Failed to connect: " . mysqli\_connect\_error();

}

//used to hash the password

function pwhash($password){

$hash=password\_hash($password,PASSWORD\_DEFAULT);

return $hash;

}

//used to check the password

function pwcheck($password,$hashed){

$hashcheck=password\_verify($password,$hashed);

return $hashcheck;

}

//used to check the login of the user, if no one is logged in, login screen shown.

function checklogin(){

if($\_SESSION["userid"]==null){

header('Location:login.php');

}

}

?>

######################### LOGIN.PHP ##############################

<!DOCTYPE html>

<html lang="en"><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta name="description" content="Website Scraper">

<meta name="author" content="Group H">

<link rel="icon" href="favicon.ico">

<title>Hotel review scraper</title>

<!-- Bootstrap core CSS -->

<link href="./css/bootstrap.min.css" rel="stylesheet">

<!-- Custom styles for this template -->

<link href="./css/style.css" rel="stylesheet">

<!-- Just for debugging purposes. Don't actually copy these 2 lines! -->

<!--[if lt IE 9]><script src="../../js/ie8-responsive-file-warning.js"></script><![endif]-->

<script src="./js/ie-emulation-modes-warning.js"></script><style type="text/css"></style>

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

</head>

<body>

<nav class="navbar navbar-inverse navbar-fixed-top">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand">Group H - Review Scraper</a>

</div>

<div id="navbar" class="collapse navbar-collapse">

<ul class="nav navbar-nav">

<li class="disabled"><a>Introduction</a></li>

<li class="disabled"><a>Scrape!</a></li>

<li class="disabled"><a>Settings</a></li>

<li class="disabled"><a>Project Info</a></li>

<li class="disabled"><a>Licensing</a></li>

</ul>

</div>

</div>

</nav>

<div class="container">

<div class="starter-template">

<h1>Login</h1>

<p class="lead">Please login to use this system:</p>

<form role="form" action='functions.php' method='post'>

<div class="form-group">

<label for="email">Email:</label>

<input type="email" class="form-control" name="email" placeholder="Enter email">

</div>

<div class="form-group">

<label for="pwd">Password:</label>

<input type="password" class="form-control" name="password" placeholder="Enter password">

</div>

<button type="login" name="login" class="btn btn-default">Login</button>

<button type="register" name="registryform" class="btn btn-default">Not a member?</button>

</form>

<?php //error checks//

session\_start();

if (isset ($\_SESSION["error"])){

if($\_SESSION["error"]=="nomatch"){

$\_SESSION["error"]="none";

echo("<span style='color: red'>Please enter the correct password.</span>");

}

elseif($\_SESSION["error"]=="fieldcheck"){

$\_SESSION["error"]="none";

echo("<span style='color: red'>Please fill in all the fields.</span>");

}

}

?>

</div>

</div>

<!-- Bootstrap core JavaScript

================================================== -->

<!-- Placed at the end of the document so the pages load faster -->

<script src="./js/jquery.min.js"></script>

<script src="./js/bootstrap.min.js"></script>

<!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->

<script src="./js/ie10-viewport-bug-workaround.js"></script>

</body></html>

###################### SETTINGS.PHP ##############################

<!DOCTYPE html>

<?php

session\_start();

include 'constants.php';

checklogin();

?>

<html lang="en"><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta name="description" content="Website Scraper">

<meta name="author" content="Group H">

<link rel="icon" href="favicon.ico">

<title>Hotel review scraper</title>

<!-- Bootstrap core CSS -->

<link href="./css/bootstrap.min.css" rel="stylesheet">

<!-- Custom styles for this template -->

<link href="./css/style.css" rel="stylesheet">

<!-- Just for debugging purposes. Don't actually copy these 2 lines! -->

<!--[if lt IE 9]><script src="../../js/ie8-responsive-file-warning.js"></script><![endif]-->

<script src="./js/ie-emulation-modes-warning.js"></script><style type="text/css"></style>

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

<script type="text/javascript">

var hook = true;

window.onbeforeunload = function() {

if (hook) {

return "Please log out first"

}

}

function unhook() {

hook=false;

}

function bypass(){

unhook();

}

</script>

</head>

<body>

<nav class="navbar navbar-inverse navbar-fixed-top">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand">Group H - Review Scraper</a>

</div>

<div id="navbar" class="collapse navbar-collapse">

<ul class="nav navbar-nav">

<li class="inactive"><a onclick="bypass()" href="home.php">Introduction</a></li>

<li class="inactive"><a onclick="bypass()" href="scrape.php">Scrape!</a></li>

<li class="inactive"><a onclick="bypass()" href="project\_info.php">Project Info</a></li>

<li class="inactive"><a onclick="bypass()" href="license.php">Licensing</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li class="active"><a onclick="bypass()" href="settings.php"><span class="glyphicon glyphicon-cog"></span> Settings</a></li>

<li class="inactive"><a onclick="bypass()" href="logout.php"><span class="glyphicon glyphicon-log-out"></span> Logout</a></li>

</ul>

</div>

</div>

</nav>

<div class="container">

<div class="starter-template">

<h1>Settings</h1>

</div>

<div class="row">

<div class="col-sm-4">

<h3>Change Account Settings</h3>

<u><b>Change password</b></u><br><br>

<form role="form" action='functions.php' method='post'>

<div class="form-group">

<label for="pwd">Current password:</label>

<input type="password" class="form-control" name="password1" placeholder="Enter current password"><br>

<label for="pwd">New password:</label>

<input type="password" class="form-control" name="password2" placeholder="Enter new password"><br>

<label for="pwd">Confirm new password:</label>

<input type="password" class="form-control" name="password3" placeholder="Confirm new password">

</div>

<button type="update" name="changepw" onclick="bypass()" class="btn btn-default">Update Password</button>

</form>

<?php //error checks//

if (isset ($\_SESSION["error"])){

if($\_SESSION["error"]=="pwsnotthere"){

$\_SESSION["error"]="none";

echo("<span style='color: red'>Please fill in all the fields.</span>");

}

else if($\_SESSION["error"]=="oldpwmatch"){

$\_SESSION["error"]="none";

echo("<span style='color: red'>Please make sure the old password is correct.</span>");

}

else if($\_SESSION["error"]=="nopwmatch"){

$\_SESSION["error"]="none";

echo("<span style='color: red'>Please make sure the new passwords match.</span>");

}

else if($\_SESSION["error"]=="pwupdated"){

$\_SESSION["error"]="none";

echo("<span style='color: green'>Password updated.</span>");

}

}

?>

</div>

<?php

// if the admin is not logged in, normal user settings shown, these inc, above pw change, and url change

if ($\_SESSION["admin"]==false){

?>

<div class="col-sm-4">

<h3>Change Hotel Settings</h3>

<B>Your project is setup with URL:</B> <br>

<?=$\_SESSION["ScrapeURL"]; ?><br><br>

<u><b>Change TripAdvisor URL</b></u><br><br>

<form role="form" action='functions.php' method='post'>

<div class="form-group">

<label >New TripAdvisor URL:</label>

<input type="text" class="form-control" name="newurl" placeholder="Paste URL here">

</div>

<button type="update" name="changeurl" onclick="bypass()" class="btn btn-default">Change URL</button>

</form>

<?php //error checks//

if (isset ($\_SESSION["error"])){

if($\_SESSION["error"]=="invalidurl"){

$\_SESSION["error"]="none";

echo("<span style='color: red'>Please enter a valid TripAdvisor URL.</span>");

}

else if($\_SESSION["error"]=="sameurl"){

$\_SESSION["error"]="none";

echo("<span style='color: red'>Please enter a different TripAdvisor URL.</span>");

}

else if($\_SESSION["error"]=="updatedurl"){

$\_SESSION["error"]="none";

echo("<span style='color: green'>URL updated.</span>");

}

}

?>

</div>

<div class="col-sm-4">

<h3>Helpful tips about settings</h3>

<br><u><b>Passwords</b></u><br>

When changing your password, use a password that is suitably secure and memorable.

<br>

If you forget your password, please contact the site admin.

<br><br><u><b>Changing website URL</b></u><br>

When you change the url of your hotel, you must submit a valid, active URL from the current TripAdvisor website. When your URL is updated, all previously saved data will be removed.

</div>

<?php

}

// if the admin is logged in, system settings shown, these inc,pw change, any user url change and system setting change

else if ($\_SESSION["admin"]==true){

?>

<div class="col-sm-4">

<h3>Change Hotel Settings</h3>

<u><b>Change a users TripAdvisor URL</b></u><br><br>

<form role="form" action='functions.php' method='post'>

<div class="form-group">

<label >Select user via Email:</label>

<select class="form-control" name="emailchoice">

<option value='na'>Select an email</option>

<?php

$user=$\_SESSION["userid"];

$emails = mysqli\_fetch\_all(mysqli\_query($con,"SELECT ownerId,email FROM Owner WHERE ownerId>0"));

foreach($emails as $row){

echo('<option>');

echo $row[1];

echo('</option>');

}?>

</select><br>

<label >New TripAdvisor URL:</label>

<input type="text" class="form-control" name="newurl1" placeholder="Paste URL here">

</div>

<button type="update" name="changeuserurl" onclick="bypass()" class="btn btn-default">Change URL</button>

</form>

<?php //error checks//

if (isset ($\_SESSION["error"])){

if($\_SESSION["error"]=="invalidurl1"){

$\_SESSION["error"]="none";

echo("<span style='color: red'>Please enter a valid TripAdvisor URL.</span>");

}

else if($\_SESSION["error"]=="sameurl1"){

$\_SESSION["error"]="none";

echo("<span style='color: red'>Please enter a different TripAdvisor URL.</span>");

}

else if($\_SESSION["error"]=="emailselect"){

$\_SESSION["error"]="none";

echo("<span style='color: red'>Please select an email.</span>");

}

else if($\_SESSION["error"]=="updatedurl1"){

$\_SESSION["error"]="none";

echo("<span style='color: green'>URL updated.</span>");

}

}

?>

<br><br><u><b>Remove a user from the system</b></u><br><br>

<form role="form" action='functions.php' method='post'>

<div class="form-group">

<label >Select user via Email:</label>

<select class="form-control" name="emailchoice">

<option value='na'>Select an email</option>

<?php

$user=$\_SESSION["userid"];

$emails = mysqli\_fetch\_all(mysqli\_query($con,"SELECT ownerId,email FROM Owner WHERE ownerId>0"));

foreach($emails as $row){

echo('<option>');

echo $row[1];

echo('</option>');

}?>

</select><br>

<button type="update" name="removeuser" onclick="bypass()" class="btn btn-default">Remove user</button>

</form></div>

<?php //error checks//

if (isset ($\_SESSION["error"])){

if($\_SESSION["error"]=="removeduser"){

$\_SESSION["error"]="none";

echo("<span style='color: green'>User removed.</span>");

}

}

?>

</div>

<div class="col-sm-4">

<h3>Admin only settings</h3>

<u><b>Change global settings for data analysis</b></u><br><br>

<form role="form" action='functions.php' method='post'>

<div class="form-group">

<label >Loop Timeout:</label>

<input type="number" class="form-control" name="looptimeout" placeholder="loop\_timeout"><br>

<label >Sleep Time:</label>

<input type="number" class="form-control" name="sleeptime" placeholder="sleep\_time"><br>

<label >Write Quota:</label>

<input type="number" class="form-control" name="writequota" placeholder="database\_write\_quota">

<div class="checkbox">

<label><input type="checkbox" value="reset" name="reset">Reset to standard - (loop=600, sleep=5, database=5)</label>

</div>

</div>

<button type="update" name="changeadminsettings" onclick="bypass()" class="btn btn-default">Change settings</button>

</form>

<?php //error checks//

if (isset ($\_SESSION["error"])){

if($\_SESSION["error"]=="resetDBsettings"){

$\_SESSION["error"]="none";

echo("<span style='color: green'>System global settings have been reset.</span>");

}

if(strpos($\_SESSION["error"],"looptime") !== false ){

echo("<span style='color: green'>Loop time has been changed.</span><br>");

}

if(strpos($\_SESSION["error"],"sleeptime") !== false ){

echo("<span style='color: green'>Sleep time has been changed.</span><br>");

}

if(strpos($\_SESSION["error"],"writequota") !== false ){

echo("<span style='color: green'>Write quota has been changed.</span>");

}

$\_SESSION["error"]="none";

}

?>

</div>

<?php } ?>

</div>

</div>

<!-- Bootstrap core JavaScript

================================================== -->

<!-- Placed at the end of the document so the pages load faster -->

<script src="./js/jquery.min.js"></script>

<script src="./js/bootstrap.min.js"></script>

<!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->

<script src="./js/ie10-viewport-bug-workaround.js"></script>

</body></html>

####################### SCRAPE.PHP #################################

<!DOCTYPE html>

<?php

session\_start();

include 'constants.php';

checklogin();

?>

<html lang="en"><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta name="description" content="Website Scraper">

<meta name="author" content="Group H">

<link rel="icon" href="favicon.ico">

<title>Hotel review scraper</title>

<!-- Bootstrap core CSS -->

<link href="./css/bootstrap.min.css" rel="stylesheet">

<!-- Custom styles for this template -->

<link href="./css/style.css" rel="stylesheet">

<!-- Just for debugging purposes. Don't actually copy these 2 lines! -->

<!--[if lt IE 9]><script src="../../js/ie8-responsive-file-warning.js"></script><![endif]-->

<script src="./js/ie-emulation-modes-warning.js"></script><style type="text/css"></style>

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

<script type="text/javascript">

var hook = true;

window.onbeforeunload = function() {

if (hook) {

return "Please log out first"

}

}

function unhook() {

hook=false;

}

function bypass(){

unhook();

}

function limitReviews() {

var dfrom = document.getElementById("dfrom").value;

var dto = document.getElementById("dto").value;

var table = document.getElementById("rws");

var childPorducts = table.children;

for(var i = 1; i < childPorducts.length; i++){

var deeperChild = childPorducts[i].children;

if(dfrom<=i && dto>=i){

childPorducts[i].style.display = "";

}

else{childPorducts[i].style.display = "none";}

}

}

</script>

</head>

<body onload="limitDates()">

<nav class="navbar navbar-inverse navbar-fixed-top">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand">Group H - Review Scraper</a>

</div>

<div id="navbar" class="collapse navbar-collapse">

<ul class="nav navbar-nav">

<li class="inactive"><a onclick="bypass()" href="home.php">Introduction</a></li>

<li class="active"><a onclick="bypass()" href="scrape.php">Scrape!</a></li>

<li class="inactive"><a onclick="bypass()" href="project\_info.php">Project Info</a></li>

<li class="inactive"><a onclick="bypass()" href="license.php">Licensing</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li class="inactive"><a onclick="bypass()" href="settings.php"><span class="glyphicon glyphicon-cog"></span> Settings</a></li>

<li class="inactive"><a onclick="bypass()" href="logout.php"><span class="glyphicon glyphicon-log-out"></span> Logout</a></li>

</ul>

</div>

</div>

</nav>

<div class="container">

<div class="starter-template">

<?php

if($\_SESSION['admin'] == true){

if (isset($\_SESSION["scanview"]) && $\_SESSION["scanview"]==true){

$\_SESSION["scanview"]=false;

echo '<FORM><button type="update" name="showuserhotel" onClick="bypass(),history.go(0)" class="btn btn-default">Show Reviews for another user</button></FORM>';

$id = $\_SESSION['usersid'];

$scrapestamp = $\_SESSION['sesion'];

$scrapetime = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT start\_time FROM Sesion WHERE id='$scrapestamp'"))['start\_time'];

$revs = mysqli\_fetch\_all(mysqli\_query($con,"SELECT \* FROM ScrapedReview JOIN Project on ScrapedReview.project\_id=Project.projectId WHERE owner\_project='$id'"));

$average = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT avg(rating) from ScrapedReview JOIN Project on ScrapedReview.project\_id=Project.projectId WHERE owner\_project='$id'"))['avg(rating)'];

$recent = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT date\_published from ScrapedReview JOIN Project on ScrapedReview.project\_id=Project.projectId WHERE owner\_project='$id' ORDER BY date\_published DESC LIMIT 1;"))['date\_published'];

$oldest = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT date\_published from ScrapedReview JOIN Project on ScrapedReview.project\_id=Project.projectId WHERE owner\_project='$id' ORDER BY date\_published ASC LIMIT 1;"))['date\_published'];

$one=0;$two=0;$three=0;$four=0;$five=0;

?>

<div class = 'column'>

<div class="col-md-4"><br>

<table class="table table-hover">

<thead>

<tr>

<th> Hotel statistics </th>

<th></th>

</tr>

</thead>

<tbody>

<tr>

<td>Average rating: </td>

<td><?=number\_format((float)$average,1,'.','') ?> </td>

</tr><tr>

<td>Reviews scraped: </td>

<td><?=count($revs)?></td>

</tr><tr>

<td>Scrape timestamp: </td>

<td><?=date('H:i:s d-m-Y', strtotime($scrapetime)-7200)?> </td>

</tr><tr>

<td>Most recent review: </td>

<td><?=$recent?> </td>

</tr><tr>

<td>Oldest review scraped: </td>

<td><?=$oldest?> </td>

</tr>

</tbody>

</table>

Rating percentage:

<div id="donutchart" style="width: 350px; height: 350px;"></div>

Rating count:

<div id="bottom\_x\_div" style="width: 350px; height: 350px;"></div>

</div>

</div>

<div class='column'>

<div class="col-md-8"><br>

<table class="table table-hover">

<thead>

<tr>

<th> Review title </th>

<th> Rating </th>

<th> Content </th>

<th> Date </th>

</tr>

</thead>

<tbody>

<tr>

<?php

foreach ($revs as $row){

echo('<tr>');

echo('<td>'. $row[8] .'</td>');

echo('<td>'. $row[4] .'</td>');

echo('<td>'. $row[3] .'</td>');

echo('<td>');

echo( date('d-m-Y',strtotime($row[7])));

echo('</td>');

echo('</tr>');

//get data for charts

if($row[4]==1){$one++;}

else if($row[4]==2){$two++;}

else if($row[4]==3){$three++;}

else if($row[4]==4){$four++;}

else if($row[4]==5){$five++;}

}

?>

</tr>

</tbody>

</table>

</div>

</div>

<?php

}

else{?>

<h3>Select a user to view their hotel stats</h3>

<form role="form" action='functions.php' method='post'>

<div class="form-group">

<label >Select user via Email:</label>

<select class="form-control" name="emailchoice">

<option value='na'>Select an email</option>

<?php

$user=$\_SESSION["userid"];

$emails = mysqli\_fetch\_all(mysqli\_query($con,"SELECT ownerId,email FROM Owner WHERE ownerId>0"));

foreach($emails as $row){

echo('<option>');

echo $row[1];

echo('</option>');

}?>

</select><br>

<button type="update" name="showuserhotel" onclick="bypass()" class="btn btn-default">Show Reviews for user</button>

</form></div>

<?php

}}

else{

if(isset($\_SESSION['loading']) && $\_SESSION['loading'] == true && $\_SESSION['admin'] == false){header('Location:loading.php');}

if(isset($\_SESSION["scanview"])==false && $\_SESSION['admin']==false){?>

<h1>Scrape!</h1>

<form class="form-horizontal" action='functions.php' method='post'>

<fieldset>

<legend>Do you wish to use older data, or to refresh the data in the database?</legend>

<?php

$userid = $\_SESSION["userid"];

$seshid = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT sesion\_id FROM SesionHasProject JOIN Project ON SesionHasProject.project\_id = Project.projectId WHERE owner\_project='$userid'")) ['sesion\_id'];

$stime = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT start\_time FROM Sesion WHERE id='$seshid'")) ['start\_time'];

$date = strtotime($stime);

if($date == strtotime("02:00:00 01-01-1970")){

echo("You have never used the system, click below to begin your first scan")?>

<br>

<div class="control-group">

<div class="controls">

<button id="button2id" name="new" class="btn btn-danger" onclick="bypass()">First time scan</button>

</div>

</div>

<?php }

else{

echo("<br>The Database was last updated at: <br>");

echo date('H:i:s d-m-Y', ($date -7200));

?><br>

<div class="control-group">

<div class="controls">

<button id="button1id" name="old" class="btn btn-success" onclick="bypass()">Use older data</button>

<button id="button2id" name="new" class="btn btn-danger" onclick="bypass()">Refresh database</button>

</div>

</div>

<?php } ?>

</fieldset>

</form>

<?php

}

else if($\_SESSION["scanview"]==true){

echo '<h2>Review analysis</h2>';

$id = $\_SESSION['userid'];

$scrapestamp = $\_SESSION['sesion'];

$scrapetime = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT start\_time FROM Sesion WHERE id='$scrapestamp'"))['start\_time'];

$revs = mysqli\_fetch\_all(mysqli\_query($con,"SELECT \* FROM ScrapedReview JOIN Project on ScrapedReview.project\_id=Project.projectId WHERE owner\_project='$id'"));

$average = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT avg(rating) from ScrapedReview JOIN Project on ScrapedReview.project\_id=Project.projectId WHERE owner\_project='$id'"))['avg(rating)'];

$recent = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT date\_published from ScrapedReview JOIN Project on ScrapedReview.project\_id=Project.projectId WHERE owner\_project='$id' ORDER BY date\_published DESC LIMIT 1;"))['date\_published'];

$oldest = mysqli\_fetch\_assoc(mysqli\_query($con,"SELECT date\_published from ScrapedReview JOIN Project on ScrapedReview.project\_id=Project.projectId WHERE owner\_project='$id' ORDER BY date\_published ASC LIMIT 1;"))['date\_published'];

$one=0;$two=0;$three=0;$four=0;$five=0;

?>

<div class = 'column'>

<div class="col-md-4"><br>

<table class="table table-hover">

<thead>

<tr>

<th> Hotel statistics </th>

<th></th>

</tr>

</thead>

<tbody>

<tr>

<td>Average rating: </td>

<td><?=number\_format((float)$average,1,'.','') ?> </td>

</tr><tr>

<td>Reviews scraped: </td>

<td><?=count($revs)?></td>

</tr><tr>

<td>Scrape timestamp: </td>

<td><?=date('H:i:s d-m-Y', strtotime($scrapetime)-7200)?> </td>

</tr><tr>

<td>Most recent review: </td>

<td><?=$recent?> </td>

</tr><tr>

<td>Oldest review scraped: </td>

<td><?=$oldest?> </td>

</tr>

</tbody>

</table>

Rating percentage:

<div id="donutchart" style="width: 350px; height: 350px;"></div>

Rating count:

<div id="bottom\_x\_div" style="width: 350px; height: 350px;"></div>

</div>

</div>

<br>

<form class="form-inline" style="margin-right: 1em;" name="pagination">

<label>Show Reviews From: </label><input type="number" style="width: 50px" id="dfrom" name="dfrom" value="0">

<label>To: </label><input type="number" style="width: 50px" id="dto" name="dto" value="50" >

<input type = "button"

onclick= "limitReviews()"

value="Limit Reviews">

</form>

<div class='column'>

<div class="col-md-8"><br>

<table class="table table-hover">

<thead>

<tr>

<th> Review title </th>

<th> Rating </th>

<th> Content </th>

<th> Date </th>

</tr>

</thead>

<tbody id="rws">

<tr>

<?php

foreach ($revs as $row){

echo('<tr>');

echo('<td>'. $row[8] .'</td>');

echo('<td>'. $row[4] .'</td>');

echo('<td>'. $row[3] .'</td>');

echo('<td>');

echo( date('d-m-Y',strtotime($row[7])));

echo('</td>');

echo('</tr>');

//get data for charts

if($row[4]==1){$one++;}

else if($row[4]==2){$two++;}

else if($row[4]==3){$three++;}

else if($row[4]==4){$four++;}

else if($row[4]==5){$five++;}

}

?>

</tr>

</tbody>

</table>

</div>

</div>

<?php } }

?>

</div>

</div>

<!-- Bootstrap core JavaScript

================================================== -->

<!-- Placed at the end of the document so the pages load faster -->

<script src="./js/jquery.min.js"></script>

<script src="./js/bootstrap.min.js"></script>

<!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->

<script src="./js/ie10-viewport-bug-workaround.js"></script>

<!-- code for charts -->

<script type="text/javascript" src="https://www.google.com/jsapi"></script>

<script type="text/javascript">

google.load("visualization", "1", {packages:["corechart"]});

google.setOnLoadCallback(drawChart);

function drawChart() {

var data = google.visualization.arrayToDataTable([

['Ratings', 'user rating'],

['1 star', <?php echo $one; ?>],

['2 star', <?php echo $two; ?>],

['3 star', <?php echo $three; ?>],

['4 star', <?php echo $four; ?>],

['5 star', <?php echo $five; ?>],

]);

var options = {

width: 400,

pieHole: 0.3,

};

var chart = new google.visualization.PieChart(document.getElementById('donutchart'));

chart.draw(data, options);

}

google.load("visualization", "1.1", {packages:["bar"]});

google.setOnLoadCallback(drawStuff);

function drawStuff() {

var data = new google.visualization.arrayToDataTable([

['Ratings', 'User ratings'],

['1 star', <?php echo $one; ?>],

['2 star', <?php echo $two; ?>],

['3 star', <?php echo $three; ?>],

['4 star', <?php echo $four; ?>],

['5 star', <?php echo $five; ?>],

]);

var options = {

width: 250,

legend: { position: 'none' },

bars: 'vertical',

axes: { x: {0: { side: 'bottom', label: 'Review count'} } },

bar: { groupWidth: "100%" }

};

var chart = new google.charts.Bar(document.getElementById('bottom\_x\_div'));

chart.draw(data, options);

};

</script>

</body></html>

1. Gantt chart in Appendix C [↑](#footnote-ref-1)
2. Only include top-level posts, not replies. [↑](#footnote-ref-2)
3. For example for a team of 6 give 6 to the person with the highest number. [↑](#footnote-ref-3)
4. You can give the same rank order to more than one team member so if all have contributed equally then for a team of 6 score 6 for all. [↑](#footnote-ref-4)