HOMESTAY ANALYTICS DASHBOARD

NURUL FADHILAH BINTI SAINAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

HOMESTAY ANALYTICS DASHBOARD

NURUL FADHILAH BINTI SAINAN

This report is submitted in partial fulfilment of the requirements for the

Bachelor of Computer Science (Computer Networking)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2017

BORANG PENGESAHAN STATUS TESIS*

JUDUL: HOMESTAY ANALYTICS DASHBOARD

SESI PENGAJIAN: SESI 2017/2018

Saya NURUL FADHILAH BINTI SAINAN
(HURUF BESAR)

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan
di Perpustakaan
Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)

__________ SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

__________ TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

____/_____ TIDAK TERHAD

(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)
Alamat tetap:34, Jalan Mewah Ria 2/6, Taman Bukit Mewah 81200, Johor Bahru, Johor.	DR. ZURINA SA'AYA
Tarikh: 18 OGOS 2017	Tarikh:29 OGOS 2017

CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM)
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

DECLARATION

I hereby declare that this project report entitled
HOMESTAY ANALYTICS DASHBOARD

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : ____________________ DATE: 18/8/2017
(NURUL FADHILAH BINTI SAINAN)

SUPERVISOR : ____________________ DATE: 29/8/2017
(DR.ZURINA SA'AYA)

ACKNOWLEDGEMENT

First and foremost, I would like to take this opportunity to thank to my supervisor of this project, Dr. Zurina binti Sa'aya for her sincere advice and guidance. She greatly inspired me to give my best in this project. When I am facing any problems about my project, she will analyse and point out the mistakes that I have done, thus provide me possible solutions to solve the problems. Also, I would like to express my special thanks to my academic advisor, Dr. Robiah Yusof for her loving and endless support throughout my degree year.

My humble thanks goes to all lecturers of Faculty of Information and Communication Technology, UTeM who had teach and helped me throughout my learning journey. An abundance of thank you goes to my helpful classmates of BITC S1G2 especially Dayana, Afiqah, Wafiy, Azlifah, Fifi, Bella, Nabilah, Asma, Syahira, Syahreen, Faiz, Syaiful, Izuddin, Afiq, Farhan and Firdaus for their kind supports throughout my study. It is honestly such a blessing for knowing all of you.

I would also love to make an honourable mention to my parents Hj Sainan and Pn. Rohana also my siblings Nurul Fadhlin, Muhammad Fahmi, Nurul Fadhlinda and Nurul Farahin for their understandings and moral supports on me in completing this project. I would not have reach this far without them.Lastly, I would like to deliver my thanks to any individual who has contributed in my final year project directly or indirectly.

ABSTRACT

Information regarding homestays on Mudah.my website needs to be gathered, measured, analysed and reported. It is to make it possible for the tourism department and city council to understand the current situation for tourism purpose and planning. Analytical dashboard is a supporting tool that can measures and visualizes homestays information that is obtained from Mudah.my website. It works by tracking the web marketplace data and mining massive amount of data to find meaning. Using this data, the analytical engine can visualizes the results so user can summarizes it at a glance. There are several questions arise when developing the system which are how to retrieve data from the website, how to store the information retrieved and what is the method to visualize the information obtained. This project consists of web crawler engine and data visualization that can retrieve advertisement information of homestays in Malaysia from Mudah.my website and visualize the data for analytical purposes. This project is developed using following steps; development of web crawler, data collection and extraction, repository development, building a platform to visualize data and implementation, validation and recommendation of project. At the end, this project develop a set of graphical presentation of homestays advertisement information. With the availability of this information, user can better monitor and stay updated with homestays information in just one platform and plan towards increasing the number of tourists visiting Malaysia that will stay at homestays during their visit and ultimately improve Malaysia tourism ranking. Therefore, this dashboard is a powerful tool for tourism department and city council to assist them for decision making and promoting.

ABSTRAK

Maklumat mengenai rumah penginapan di laman web Mudah.my perlu dikumpulkan, diukur, dianalisis dan dilaporkan. Ia adalah untuk membolehkan jabatan pelancongan dan majlis bandar memahami keadaan semasa bagi tujuan dan perancangan pelancongan. Papan pemuka analitik adalah alat sokongan yang dapat mengukur dan memvisualisasikan maklumat rumah penginapan yang diperoleh dari laman web Mudah.my. Ia berfungsi dengan menjejaki data pasaran pasaran dan perlombongan jumlah besar data untuk mencari makna. Dengan menggunakan data ini, enjin analitik boleh memvisualisasikan keputusan supaya pengguna dapat meringkaskannya dengan pantas. Terdapat beberapa soalan yang timbul ketika membangunkan sistem yang bagaimana untuk mendapatkan data dari laman web, cara menyimpan maklumat yang diambil dan apakah kaedah untuk memvisualisasikan maklumat yang diperoleh. Projek ini terdiri daripada enjin 'crawler' web dan visualisasi data yang dapat memperoleh maklumat iklan rumah penginapan di Malaysia dari laman web Mudah.my dan memvisualisasikan data untuk tujuan analisis. Projek ini dibangunkan dengan menggunakan langkah berikut; Pembangunan 'crawler' web, pengumpulan data dan pengekstrakan, pembangunan repositori, membina platform untuk memvisualisasikan data dan pelaksanaan, pengesahan dan cadangan projek. Pada akhirnya, projek ini membangun satu set persembahan graf maklumat iklan rumah penginapan. Dengan adanya maklumat ini, pengguna boleh memantau dan mengemas kini maklumat rumah penginapan dengan lebih baik dalam satu platform dan merancang untuk meningkatkan jumlah pelancong yang melawat Malaysia yang akan tinggal di rumah penginapan semasa lawatan mereka dan akhirnya meningkatkan kedudukan pelancongan Malaysia. Oleh itu, papan pemuka ini adalah alat yang berkuasa untuk jabatan pelancongan dan dewan bandar untuk membantu mereka membuat keputusan dan mempromosi.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Introduction

Information regarding homestay on public marketplace website needs to be gathered, measured, analysed and reported. It is to make it possible for the tourism department and city council to understand the current situation for tourism purpose and planning. Analytical dashboard is a supporting tool that can measures and visualizes homestays information that is obtained from this online public marketplace. It works by tracking the web marketplace data and mining massive amount of data to find meaning. Using this data, the analytical engine can visualizes the results so user can summarizes it at a glance. Specifically in this project, we are focusing on homestay information that are advertised on Mudah.my. Mudah.my is an online marketplace that focuses primarily to the Malaysian market. In particular, this project consists of web crawler engine and data visualization that can retrieve advertisement information of homestays in Malaysia from Mudah.my website and visualize the data for analytical purposes. At the end, this project develop a set of graphical presentation of homestays advertisement information. With the availability of this information, user can better monitor and stay updated with homestays information in just one platform and plan towards increasing the number of tourists visiting Malaysia that will stay at homestays during their visit and ultimately improve Malaysia tourism ranking. Therefore, this dashboard is a powerful tool for tourism department and city council to assist them for decision making and promoting.

1.2 Problem Statement (PS)

Table 1. 1: Problem Statement

No	Project Problem
PS1	Homestay information from online public marketplace need to be gathered and reported on temporal basis.

1.3 Project Question (PQ)

Table 1. 2: Project Question

PS	PQ	Project Question
PS1	PQ1	How to crawl the online public marketplace to extract homestay information?
	PQ2	How to store the information retrieved from online public marketplace?
	PQ3	How to summarize and visualize the data gathered?

1.4 Project Objective (PO)

Table 1. 3: Project Objective

PS	PQ	PO	Project Objective
PS1	PQ1	PO1	To develop a web crawler to retrieve information of homestay from online public marketplace.
	PQ2	PO2	To extract information from the retrieved data and build a local repository to store the information.
	PQ3	PO3	To develop a platform to summarize and visualize homestay information.

1.5 Project Scope (PS)

i. Web crawler
   - Develop a web crawler using Python scripting.

ii. Local repository
   - Build a local repository as a database to store the data retrieved.

iii. Dashboard
   - Develop a dashboard to visualize the data gathered in a structural manner.

iv. Source of information
   - The information for the project will be retrieved and extracted from online public marketplace Mudah.my.

1.6 Project Contribution (PC)

Table 1. 4: Project Contribution

PS	PQ	PO	PC	Project Contribution
PS1	PQ1	PO1	PC1	A crawler engine to crawl data from online public marketplace.
	PQ2	PO2	PC2	Homestay repository.
	PQ3	PO3	PC3	Reporting tools for homestay information.

1.7 Thesis Organization

i. Chapter 1: Introduction

This chapter will be discussing about the project background including introduction to project, project problem statement, project questions, project objectives, project scope and project contributions.

ii. Chapter 2: Literature Review

This chapter will be discussing about literature review of the project. This chapter will provide related works, critical review of current problems and justification and proposed a solution to the issue discussed.

iii. Chapter 3: Project Methodology

This chapter will discuss about the methodology used for development of the complete system. The project methodology will be divided into phases. Project milestones and Gantt chart also will be provided.

iv. Chapter 4: Analysis and Design

This chapter will discuss about the requirement analysis including software and hardware requirement, high-level and detailed design including software and physical database design.

v. Chapter 5: Implementation

This chapter will be discussing about the software development environment setup, software configuration management and also implementation status.

vi. Chapter 6: Testing

This chapter will be discussing about the project test plan including test organization, test environment and test schedule , test strategy including classes of test, test design including test description and test data and lastly test results and analysis.

vii. Chapter 7: Conclusion

This chapter consists of summarization of project, project contribution, project limitation and future works. Strengths and weaknesses will also be stated later.

1.8 Conclusion

Web crawler is designed to retrieve data from website and the information will be kept in a local repository. Crawler works by tracking the web marketplace data and mining massive amount of data to find meaning. It is crucial to follow the standards in web crawler to avoid denial of service attack.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

Web crawler already exists since the biggest information space, the World Wide Web (www) become prevalence. Since then, web crawler is often depicted as an internet bot was created to retrieve a wide range of data from the internet. From that point forward, a wide assortment of theories and methodology of web crawlers are accessible to clarify the operation and functionalities of web crawler itself. Be that as it may, only a few topics will be covered and discussed in this chapter.

2.2 Related Work

2.2.1 Overview of Web crawler

A web crawler or spider, is an Internet bot that browse the World Wide Web (WWW) systematically and efficiently commonly with the purpose of Web indexing (web spidering). Web search engines and some other sites use web crawling to keep their web content updated. Web crawlers can duplicate all the pages they visit for later handling and processing by a search engine which lists and indexes the download pages so that users can search with more proficiently. Crawlers expend resources on the systems they visit and frequently visit without approval. Issues of load, schedule and "politeness" become an issue when large number of pages they accessed. Mechanism exists for open sites not wishing to be crawled to makes this known to crawling agent. As example, including a robots.txt file can requests bots to index just parts of a site or nothing at all. As the quantity of pages on the web is extremely large, make even the largest crawlers miss of making a complete index. Thus, making search engines bad at giving relevant search results in the early years of World Wide Web. However, this have been improved significantly by present day web crawlers. Crawlers can also validate HTML code and hyperlinks.

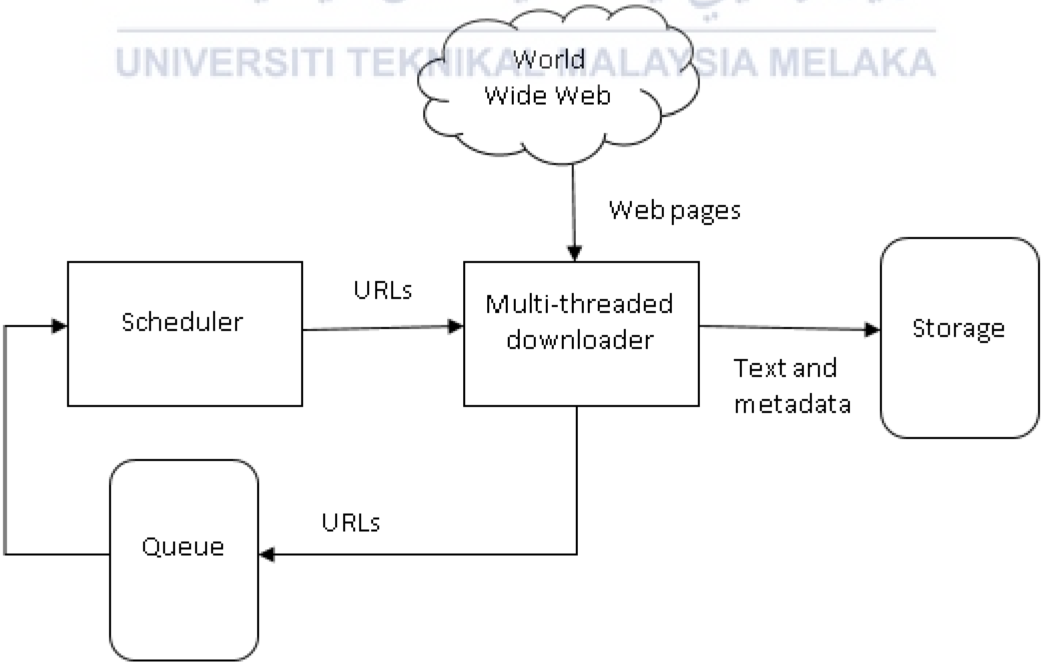


Figure 2. 1: Web crawling architecture

2.2.2 Web Crawler Properties

First, scalability. Since the web is enormous and continually growing and developing, a crawler ought to scale linearly with the quantity of agent-machines that are added to the system. Second, speed. since we want to crawl large number of pages daily, we need to employ a number of techniques such as simultaneous connections, data compression, DNS caching, minimize disk seeks and so much more in order to accomplish specific speed we need. Third, politeness. It is essential when we need to load large number of pages from the web to avoid overloading on one particular server. It is suggested to wait 10 times the time it took to download the last page. Fourth, quality. The data retrieved must be guaranteed high quality. A reliable method must be used when retrieving the information to avoid unwanted information. Fifth, robustness. Web servers might have spider trap that trap the crawler into infinite loop. Sixth, performance. System resources need to be used efficiently without producing huge traffic load at a time. Seventh, freshness where the information retrieved on a timely basis to keep track any information changes. Eighth, extensible where the crawler must be created so that be able to cope with any updates on the web site and lastly distributed where the web crawlers must support all platforms to be executed

2.2.3 Web Crawling Policy

There are four policies that become factors of a behaviour of a web crawler which are selection policy that states which pages to download, re-visit policy that states when to check for changes to the pages, politeness policy that states how to avoid from overloading web sites and parallelization policy that states how to coordinate distributed web crawlers.

2.2.4 Type of Web Crawler

There are several strategies being employed in web crawling which are focused web crawling, incremental crawling, distributed crawling and parallel crawling. In this topic our main focus is focused web crawling. Focused web crawler fetch documents that are significant to a specific topic. This homestay web crawler is built with all the properties discussed above and using the focused web crawling to fetch pages with the same URL specified.

2.3 Critical review of current problem and justification

There are many types of web crawling method exists. For this project, the system is developed a web crawler that focus on focused web crawling. The web crawler will collect information that are specific and relevant to particular topic.

According to (Verma, 2016), the breadth first crawler produce the most quality pages amid the early phases of the crawling. They hypothesized that breadth first crawler is a good crawling method since it have many links to the most important pages and those links will be found early.

According to (Edwards, Mccurley, & Tomlin, 2001), crawling using incremental web crawler strategy without making any theoretical assumptions about the rate of change of pages. But the crawler generate more reliable data by using information gleaned from actual cycles.

According to (Mahar & Jha, 2015), hidden web crawler strategies retrieves high quality data. Hence, an indexing technique must be implemented to index the high quality more efficiently.

According to (Yang Jian You, 2012), The online system is developed to ease booking process to user, develop a searching and database synchronization.

2.4 Proposed Solution/further project

Homestay crawler will be developed to retrieve homestay information from web marketplace Mudah.my using Python scripting. The crawler will be scheduled using task scheduler to crawl information in timely basis (daily). Beautiful soup HTML parser will parse the requested content. The data wanted will be analysed and retrieved by using the navigating features provided by beautiful soup. Beautiful soup will navigate the parse tree and the data obtained will be stored in a repository that have been created.

2.5 Conclusion

In this chapter, we have discussed about the properties that should have in the homestay crawler. We also have discussed the type of web crawling strategies that will be used. The project methodology and project milestone will be discussed in the next chapter.

CHAPTER III

PROJECT METHODOLOGY

3.1 Introduction

The project methodology is a base to plan, arrange, control and structure the life cycle of system development. Homestays Retrieval Dashboard is a system that is developed to retrieve information regarding homestays in Malaysia from Mudah.my web site and then the information retrieved will be extracted to be stored in a local repository. The waterfall model will be implemented in this project. This project is developed using following steps; development of web crawler, data collection and extraction, repository development, building a platform to visualize data and implementation, validation and recommendation of project.

3.2 Methodology

This project is developed step by step to avoid overlapping of phase. The five phases involved in this project is illustrated in Figure 3.1. Phase 1 is development of web crawler. A web crawler is developed to crawl homestays information specifically. A web marketplace such as Mudah.my. Phase 2 is data collection and extraction. Crawling process will be performed to collect and gather homestay information. Then this information will be extracted to get the structured data related to homestay. This structured data be analysed to filter only relevant information and to avoid data duplications. Phase 3 is repository development. This phase includes performing indexing on the data retrieved and develop a local repository to store the data that have been retrieved and gathered. Phase 4 is building a platform to visualize data. such as dashboard will be developed to summarize and visualize all the data that have been gathered. Phase 5 is implementation and testing. In this phase, the system will be tested and implemented. The testing phase is very important as it ensure that the project has completed and met the requirements of the project. Testing also whether the system is validated to real world environment and situation thus make recommendation for further beneficial usage.

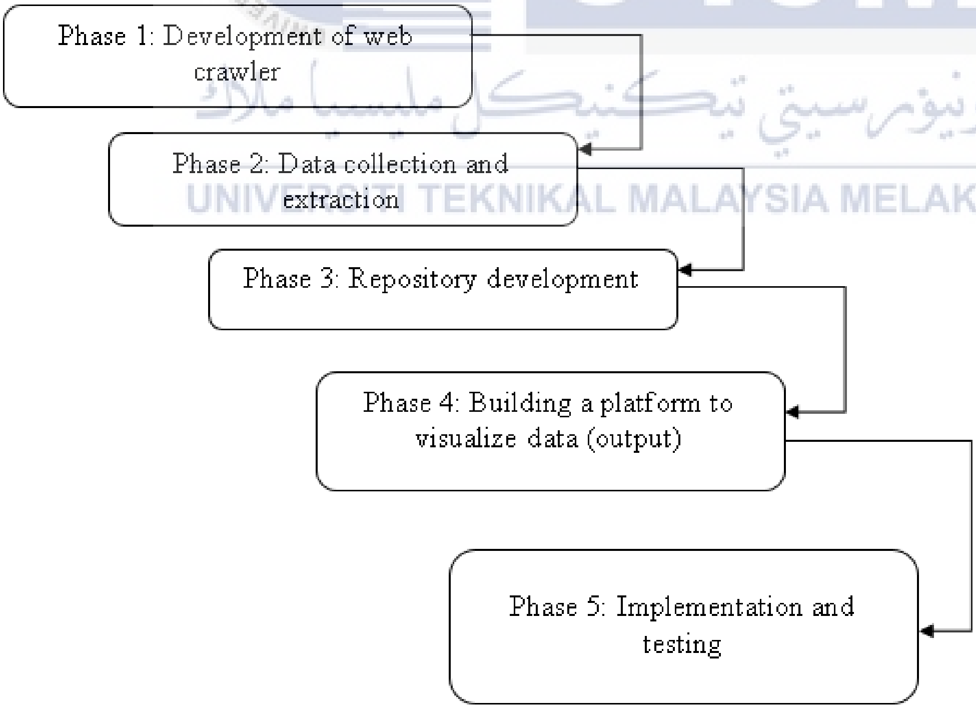

Phase 1: Development of web crawler
Phase 2: Data collection and extraction
Phase 3: Repository development
Phase 4: Building a platform to visualize data (output)
Phase 5: Implementation and testing

Figure 3. 1: Project Methodology

3.3 Project Milestones

Table 3. 1: Project Milestone

WEEK	START	END	ACTIVITY
1 (meeting 1)	13/2/2017	17/2/2017	Proposal PSM: Discussion
			Proposal assessment & verification
2	20/2/2017	24/2/2017	Proposal correction/improvement
			List of supervisor/title
3 (meeting 2)	27/2/2017	3/3/2017	Proposal presentation & submission via PSM online system
			Chapter 1(System development begins)
4	6/3/2017	10/3/2017	Chapter 1
			Chapter 2
5	13/3/2017	17/3/2017	Chapter2
6 (meeting 3)	20/3/2017	24/3/2017	Chapter 2
			Chapter 3
			Student status
7	27/3/2017	31/3/2017	Chapter 3
			Chapter 4
8	3/4/2017	7/4/2017	MID SEMESTER BREAK
9	10/4/2017	14/4/2017	Chapter 4
			Project demo
10 (meeting 4)	17/4/2017	21/4/2017	Chapter 4
			Project demo
			Student status
11 (demonstration)	24/4/2017	28/4/2017	Project demo
			Determination of student status(continue/withdraw)
12	1/5/2017	5/5/2017	Project demo
			PSM1 report

WEEK	START	END	ACTIVITY
13 (meeting 5)	8/5/2017	12/5/2017	Project demo
			PSM1 report
			PSM1 showcase poster submission
14	15/5/2017	19/5/2017	Project demo
			Submission of the PSM1 report onto PSM e-Repository online system
15 (PSM1 showcase)	22/5/2017	26/5/2017	PSM1 showcase
16	29/5/2017	2/6/2017	Revision week -correction on the draft report -Submit PSM1 logbook to PSM online system
			Submission of overall marks to PSM/PD committee
17&18	5/6/2017	18/6/2017	Final examination weeks

3.3.1 Gantt Chart

Table 3. 2: Project Gantt chart

Activity	Week														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Submit proposal to Supervisor	■														
Discussion/ Enhancement Proposal		■													
Chapter 1			■												
Chapter 1 and 2 (Demo)				■											
Chapter 2					■										
Chapter 2 and 3 (Demo)						■									
Chapter 3 and 4 (Demo)							■	■							
Chapter 4 (Demo) and 5									■	■	■	■			
PSM Report													■	■	
Final Presentation															■

3.4 Conclusion

This project will be carried out in 15 weeks. Throughout the duration, the project is divided into 5 phases following the waterfall methodology as mentioned earlier. The system development begins alongside with chapter 1 after the project problem statement and project objectives is analysed and defined until the end of system development. The project milestone and project Gantt chart act as a guideline to accomplish progress at each time to keep the system development in track. The next chapter will be discussing about the project analysis and design.

CHAPTER IV

ANALYSIS AND DESIGN

4.1 Introduction

In this chapter, the design phase will be discussed in more detail. Hardware and software requirement will be explained deeply. The system component and architecture will be defined as to assist in system development. Later, the system will be set up following the flow chart as guideline.

4.2 Problem Analysis

Currently, tourism department of city council does not keep any records of current trend and updates of homestays information. This will be a constraint for them to keep track of the latest homestays data in details. This could lead to some issues such as bad tourism planning in Malaysia. This crawler engine is developed to reduce these issues where homestays information can be retrieved and updated on a timely basis. The proposed web crawler will be able to retrieve the homestays data through online web marketplace called Mudah.my.

4.3 Requirement Analysis

4.3.1 Data Requirement

The data requirement will indicate the data specifications that stored in the repository of system by the crawler.

Mudah repository

The mudah repository with a table name of homestays held the information of homestays obtained from Mudah.my web site.

Table 4. 1: Mudah repository

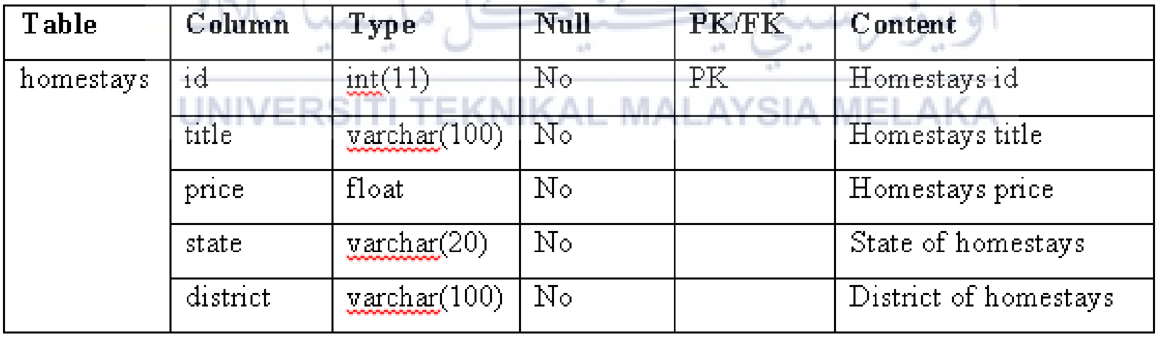


Table	Column	Type	Null	PK/FK	Content
homestays	id	int(11)	No	PK	Homestays id
	title	varchar(100)	No		Homestays title
	price	float	No		Homestays price
	state	varchar(20)	No		State of homestays
	district	varchar(100)	No		District of homestays

4.3.2 Functional Requirement

The crawler works in one mode only which is triggered by a task scheduler to be crawled.

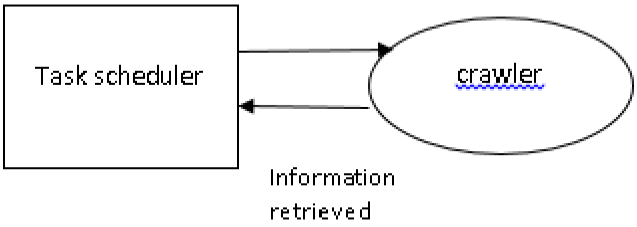


Figure 4. 1: Context Level Diagram of Crawler

From figure above, the process of crawling starts when the task scheduler trigger the crawler at a timely basis (daily).

4.3.3 Non-functional Requirement

In this project, there are several non-functional requirements included. One of the most important requirements is freshness. We are certain that web marketplace updates their websites information every once in a while to keep the data freshness maximally. We can keep the freshness of information in our repository by retrieving the information from web marketplace in a timely basis. Other than that, a sufficient network bandwidth is also a non-functional requirement to host the web crawler. It is significant to ensure the performance of the crawler. An adequate network bandwidth can deal with the traffic produced by the crawler itself to eliminate any interference amid the crawling process.

The crawler also must be scalable towards the development in the number of homestays. It must have the capacity to retrieve huge amount of information as the number of homestays grows. A few websites send spider traps that could prompt infinite loops inside it by a crawler. The robustness of the crawler is important to counteract it to be trapped. The politeness of the crawler is another requirement to prevent the crawler's IP address from getting blacklisted that could make future issues to the network users.

4.3.4 Other Requirement

Will be discussed about other requirements such as software and hardware requirement.

4.3.4.1 Software Requirement

Table 4. 2: Software Requirement

Software	**Description**
PyCharm 2016.3.2	PyCharm is an Integrated Development Environment (IDE) used in computer programming, specifically for the Python language. It provides code analysis, a graphical debugger and an integrated unit tester. PyCharm is a cross-platform with Windows, macOS and Linux versions.
Macromedia Dreamweaver 8.0	A proprietary web development tool developed by Adobe Systems. Use as a platform to develop dashboard as reporting tool.
Xampp Control Panel v3.2.2	Xampp server is an open source web server solution. It is supported by all major platforms including windows.
Apache Web Server 2.4.25	Apache provides important features such as virtual hosting and compression features to reduce the web server load. Major server-side programming language such as Python and PHP is supported by Apache web server.
MySQL 15.1	MySQL is a relational database management system. It will be used in this project to build the local repository system to store the information of homestay in Malaysia retrieved by the crawler.
Python 2.7.13	Python is a great object-oriented, interpreted, and interactive programming language. Use to do the scripting for homestay crawler.
PHP 7.1.1	PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server. Use to do scripting for statistics dashboard website.

Software	Description
Javascript	It has an API that works with array, texts and regular expressions. Use to create an array to insert data into table.
Windows OS	Windows OS work as a server platform to run task scheduler.

4.3.4.2 Hardware Requirement

Table 4. 3 Hardware Requirement

	Minimum	Recommended
Processor	1 GHz dual-core 64-bit processor and above	2 GHz quad-core 64-bit processor and above
Ram	1Gb or more	4Gb or more
Disk space	Minimum free space of 1Gb	Free space of 5Gb or more
Internet	WAN connection with a minimum bandwidth of 5mbps	WAN connection with a bandwidth of 10mbps or more.

4.4 High-Level Design

4.4.1 System Architecture

The system architecture below shows that the crawling engine is able to get the web pages from the internet to retrieve the information contained inside it. Then, scheduler will play its role to schedule. The publication data will be updated on a timely basis (daily) to keep the database updated. The information kept in the database will be processed and arranged into structured and then visualized n dashboard.

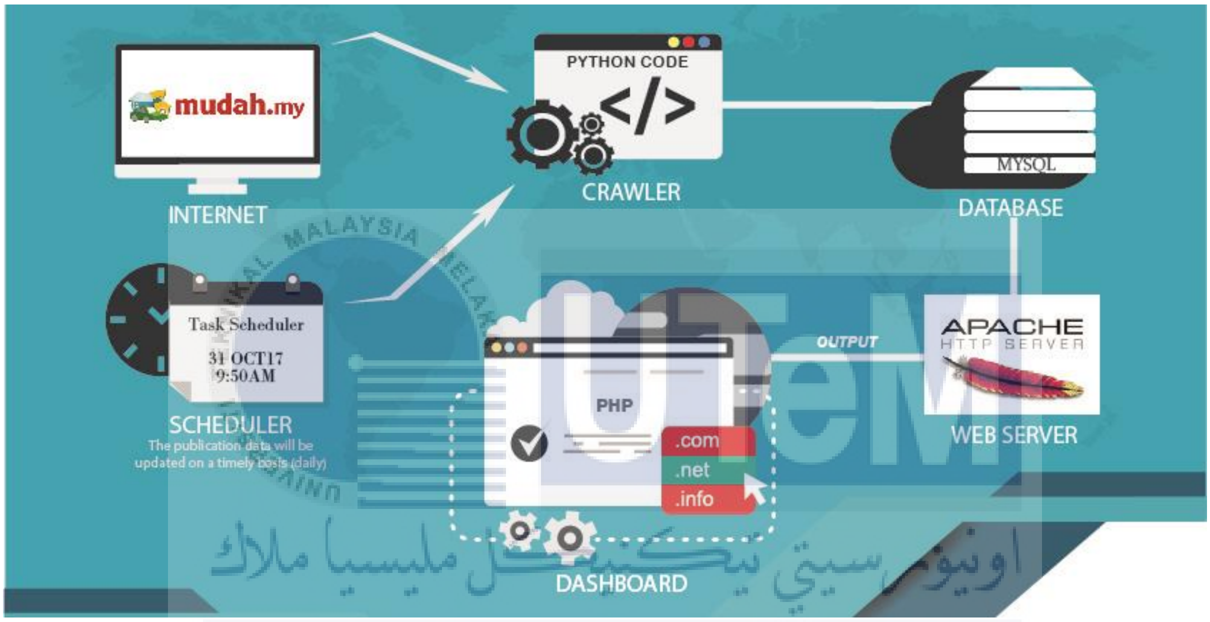


Figure 4. 2 System Architecture

4.4.2 User Interface Design

4.4.2.1 Navigation Design

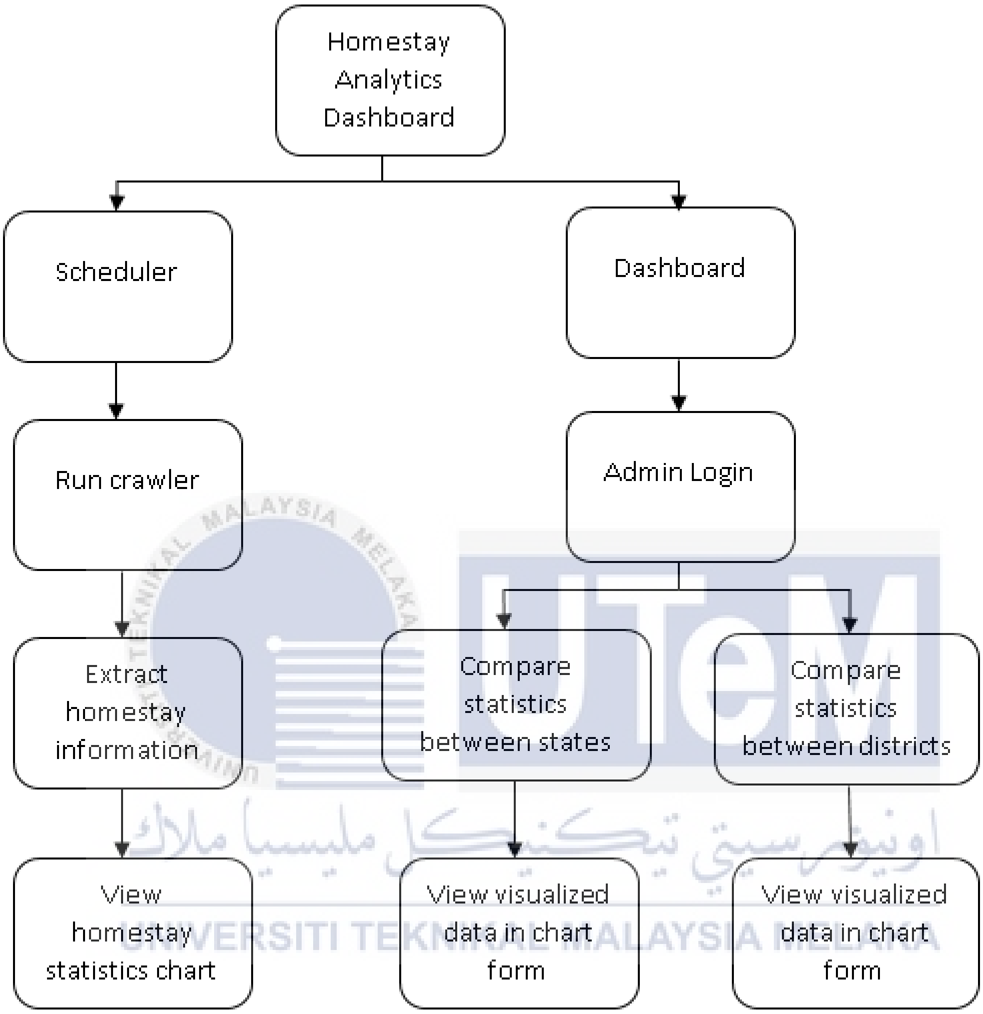

Homestay Analytics Dashboard
Scheduler
Dashboard
Run crawler
Admin Login
Extract homestay information
Compare statistics between states
Compare statistics between districts
View homestay statistics chart
View visualized data in chart form
View visualized data in chart form

Figure 4. 3 Navigation Design of Homestay Analytics Dashboard

4.4.2.2 Input Design

Table 4. 4 Table of Input Design

Interface	**Input**	**Type**	**Validation Rule**
Scheduler	-time -interval -username -password	Text box	Not null
Run homestay crawler	-crawler arguments	Run crawler	Not null
Crawl information of homestays	-state -district	Text box	Not null
Statistics comparison between states	-state	Text box	Not null
Statistics comparison between districts	-district	Text box	Not null

4.4.2.3 Output Design

The output design specifies the expected output of the system. The homestay analytics is expected to have an output to provide visualized data in form of charts and tables.

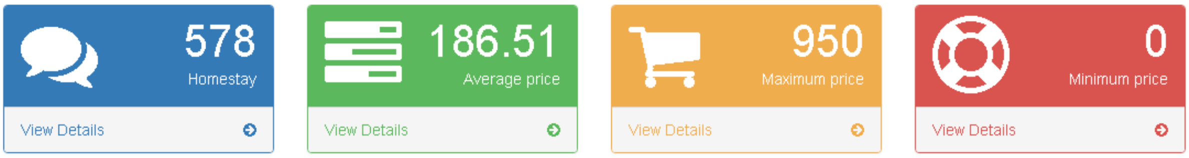


Figure 4. 4 Example data visualization of Homestay Statistics in Malaysia

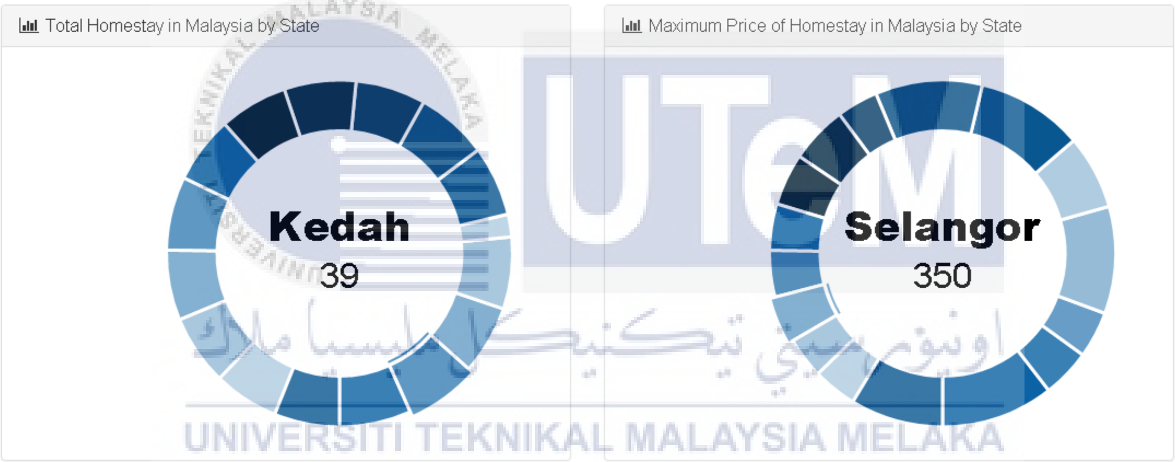

Total Homestay in Malaysia by State
Kedah
39
Maximum Price of Homestay in Malaysia by State
Selangor
350

Figure 4. 5 Example of comparison of homestay data

4.4.3 Database Design

Database design will characterize the design of repository of the system to keep the information gathered.

4.4.3.1 Conceptual and Logical Database Design

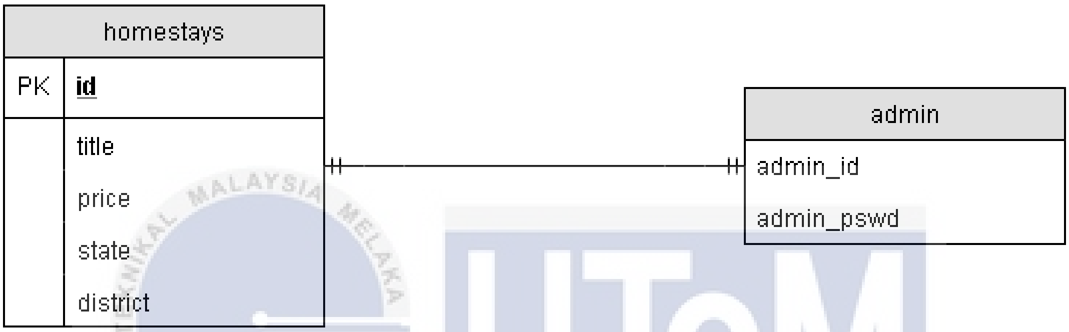


Figure 4. 6 ER Diagram

4.5 Detailed Design

Introducing the software design including the flowchart and pseudo code.

4.5.1 Software Design

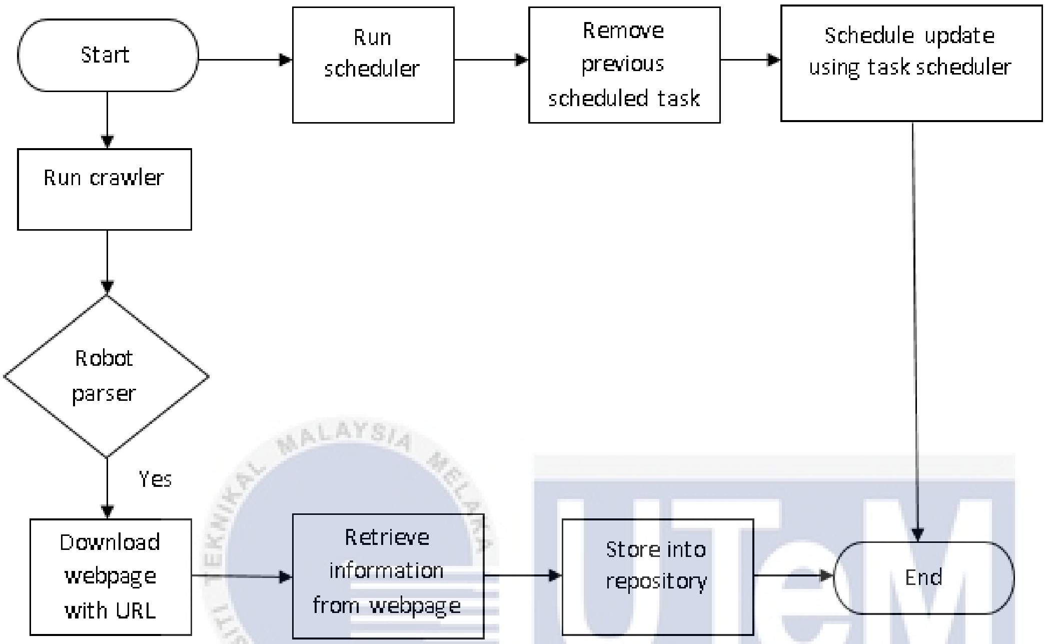

Start
Run scheduler
Remove previous scheduled task
Schedule update using task scheduler
Run crawler
Robot parser
Yes
Download webpage with URL
Retrieve information from webpage
Store into repository
End

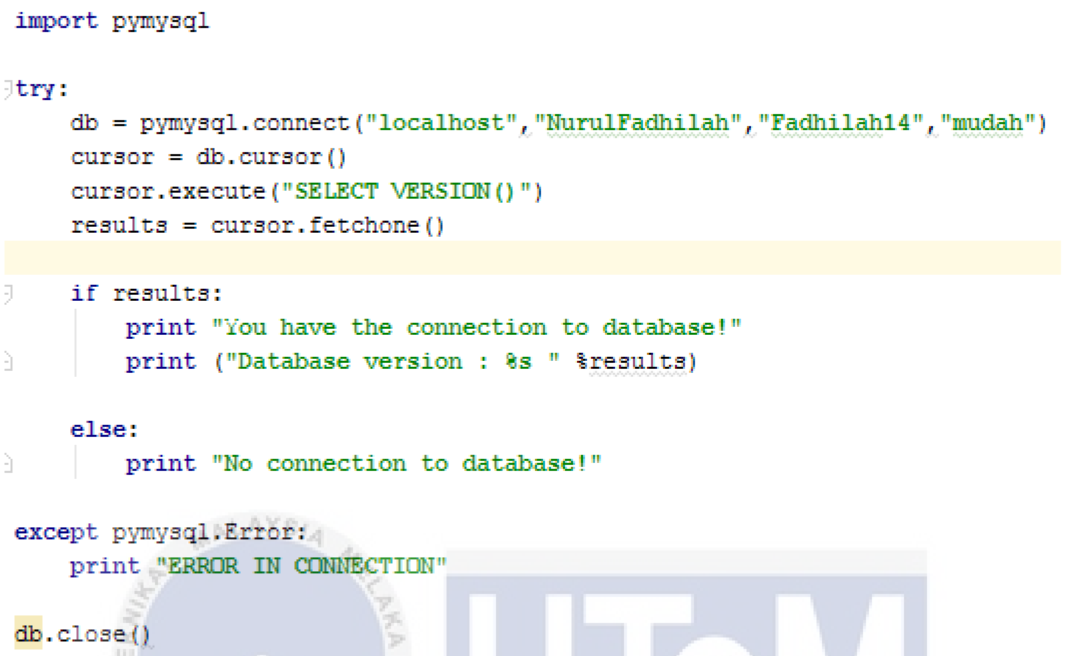
Figure 4. 7 System flowchart

4.5.2 Physical Database Design

The physical database design will be translated into SQL statements where the entities in the logical data model will be converted in the form of tables and attributes as its columns. Unique keys in each column must not be left as null.

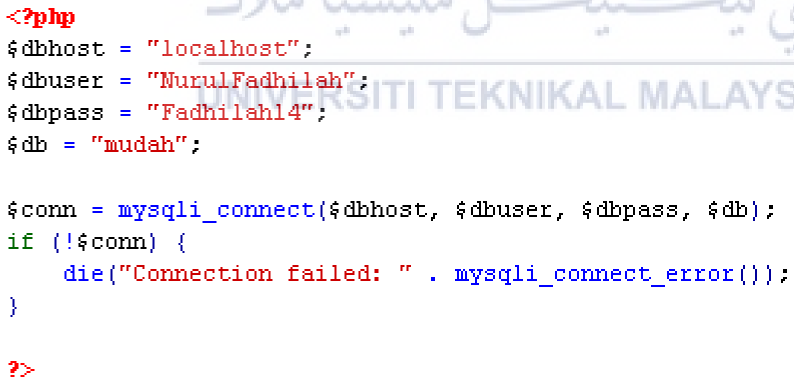
4.5.2.1 Database Connection

The database system will be installed in the similar web server that host the homestay crawler, therefore the database connection can be connected through localhost with the username of ‘NurulFadhilah’ and password ‘Fadhilah14’



```
import pymysql

try:
    db = pymysql.connect("localhost","NurulFadhilah","Fadhilah14","mudah")
    cursor = db.cursor()
    cursor.execute("SELECT VERSION()")
    results = cursor.fetchone()

    if results:
        print "You have the connection to database!"
        print ("Database version : %s " %results)

    else:
        print "No connection to database!"

except pymysql.Error:
    print "ERROR IN CONNECTION"

db.close()
```

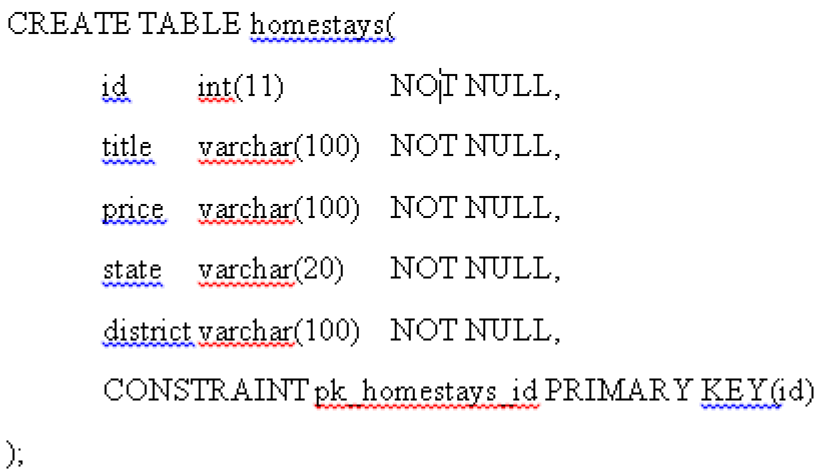
Figure 4. 8 Database Connection Using Python



```
<?php
$dbhost = "localhost";
$dbuser = "NurulFadhilah";
$dbpass = "Fadhilah14";
$db = "mudah";

$conn = mysqli_connect($dbhost, $dbuser, $dbpass, $db);
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

?>
```

Figure 4.9 Database connection using PHP

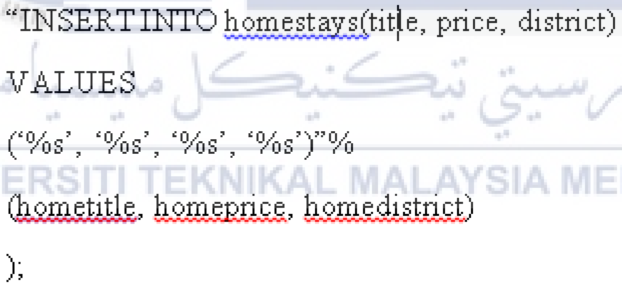
4.5.2.2 Create Homestay Table



```
CREATE TABLE homestays(
        id      int(11)         NOT NULL,
        title   varchar(100)    NOT NULL,
        price   varchar(100)    NOT NULL,
        state   varchar(20)     NOT NULL,
        district varchar(100)   NOT NULL,
        CONSTRAINT pk_homestays_id PRIMARY KEY(id)
);
```

Figure 4.10 Create homestay database

4.5.2.3 Insert Data Into Homestay Table



```
“INSERT INTO homestays(title, price, district)
VALUES
(‘%s’, ‘%s’, ‘%s’, ‘%s’)”%
(hometitle, homeprice, homedistrict)
);
```

Figure 4. 11 Insertion Data into Database

4.6 Conclusion

This chapter provides an outline to the requirement of analysis and design that must be followed in this project development. This chapter also discussed about software requirement to develop the system. Flow of the system also shown in the flowchart as a guideline to design and develop the system. In the next chapter, implementation phase where the implementation and development of the actual system will be discussed.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

Previously, phase 1 until phase 4 of project is completely done. Development of web crawler, data collection and extraction, repository development and develop a platform to visualize information. In this chapter, will discuss about implementation phase of the system. Software configuration, software setup, database connection and environment setup. This phase will make the upcoming system implementation run smoothly and efficiently.

The implementation of web crawler and Homestay Analytics Dashboard will be explained in details. Discussion includes software development setup, environment architecture, software configuration management, configuration management setup, version control procedure and implementation status of system. The implementation status of homestay crawler and dashboard will be discussed in detail.

5.2 Software Development Environment Setup

Overall, a responsive dashboard is developed that act as a reporting tool that display the information retrieved from the web using web scraping method. Hence, the requirement is to have two different development environment that shares the same repository.

For the development of crawler, the IDE used includes PyCharm with Python 2.7.13 IDLE. For the dashboard development, Dreamweaver is used as working platform with PHP. Xampp server consists of Apache http server also used to run the PHP scripting language of homestay analytics dashboard. While MySQL server used as the repository by both software developed.

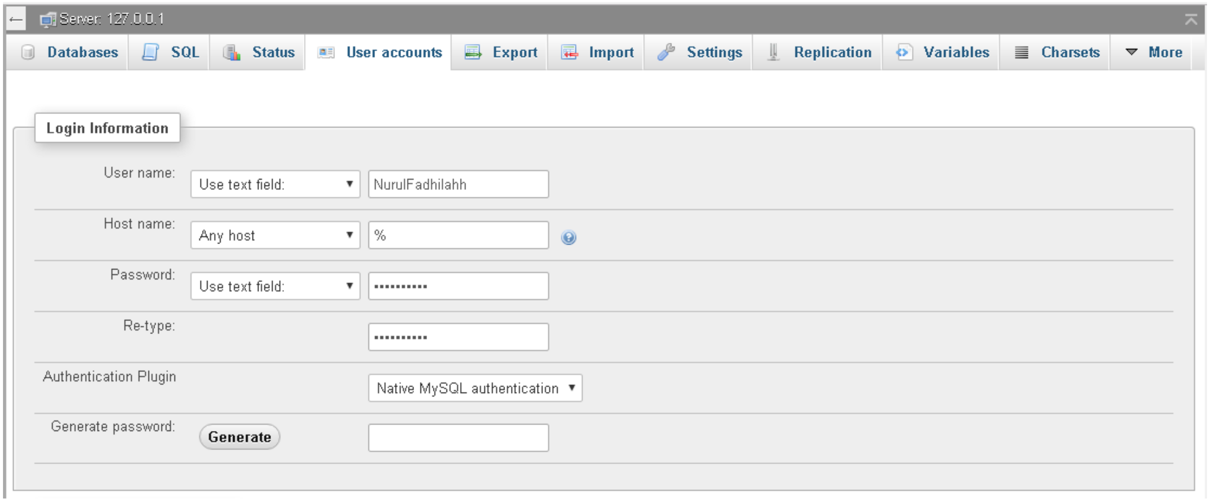
Task scheduler is created and set to update daily. A file from where the batch file is located is inserted to ensure that the specific task created will run the program daily.

5.3 Software Configuration Management

This sub topic will explained about database server setup and webserver included for the system.

5.3.1 Configuration environment setup

For the database server setup, open phpMyAdmin in the browser and login as root to add user for homestay crawler. This repository will be used by both crawler server and dashboard server.



Login Information

User name: Use text field: NurulFadhilahh

Host name: Any host %

Password: Use text field: ••••••••••

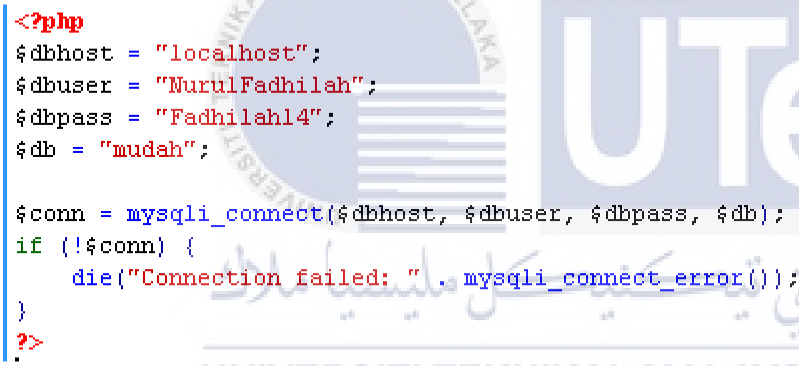
Re-type: ••••••••••

Authentication Plugin Native MySQL authentication

Generate password:

Figure 5.1: MySQL user account

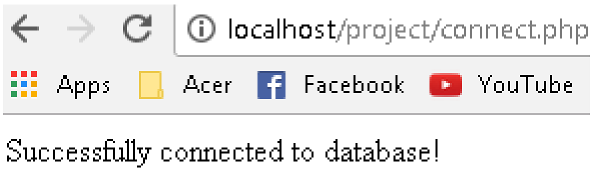
Next, create PHP scripting to establish the connection to database for dashboard.



```
<?php
$dbhost = "localhost";
$dbuser = "NurulFadhilah";
$dbpass = "Fadhilah14";
$db = "mudah";

$conn = mysqli_connect($dbhost, $dbuser, $dbpass, $db);
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
?>
```

Figure 5.2: PHP connection to database

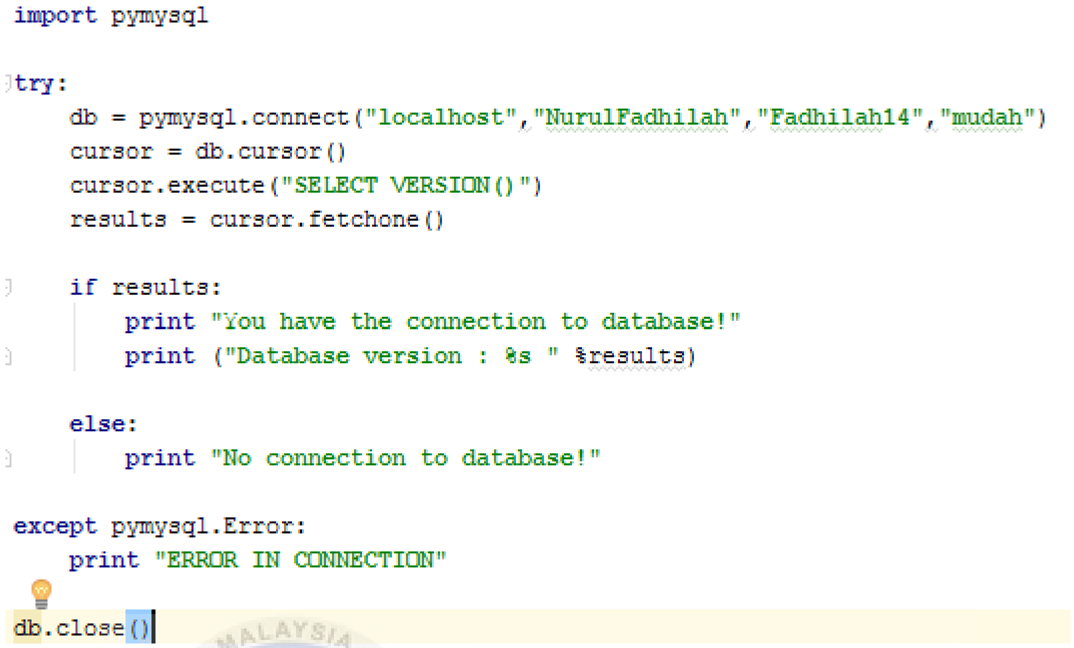
Test the database connection.



Successfully connected to database!

Figure 5.3: Output of database connection

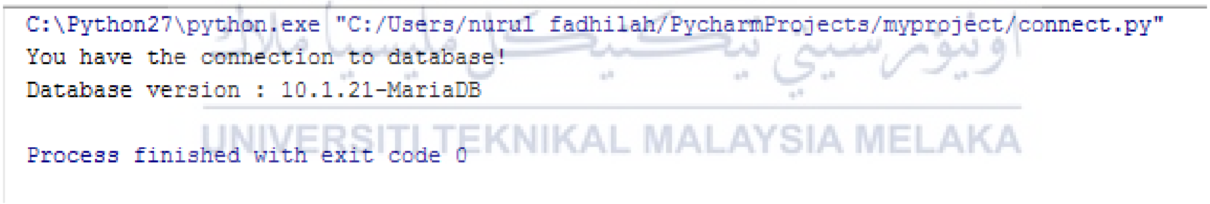
Next, create python scripting to establish the connection to database for crawler.



```
import pymysql

try:
    db = pymysql.connect("localhost","NurulFadhilah","Fadhilah14","mudah")
    cursor = db.cursor()
    cursor.execute("SELECT VERSION()")
    results = cursor.fetchone()

    if results:
        print "You have the connection to database!"
        print ("Database version : %s " %results)

    else:
        print "No connection to database!"

except pymysql.Error:
    print "ERROR IN CONNECTION"

db.close()
```

Figure 5.4: Python connection to database

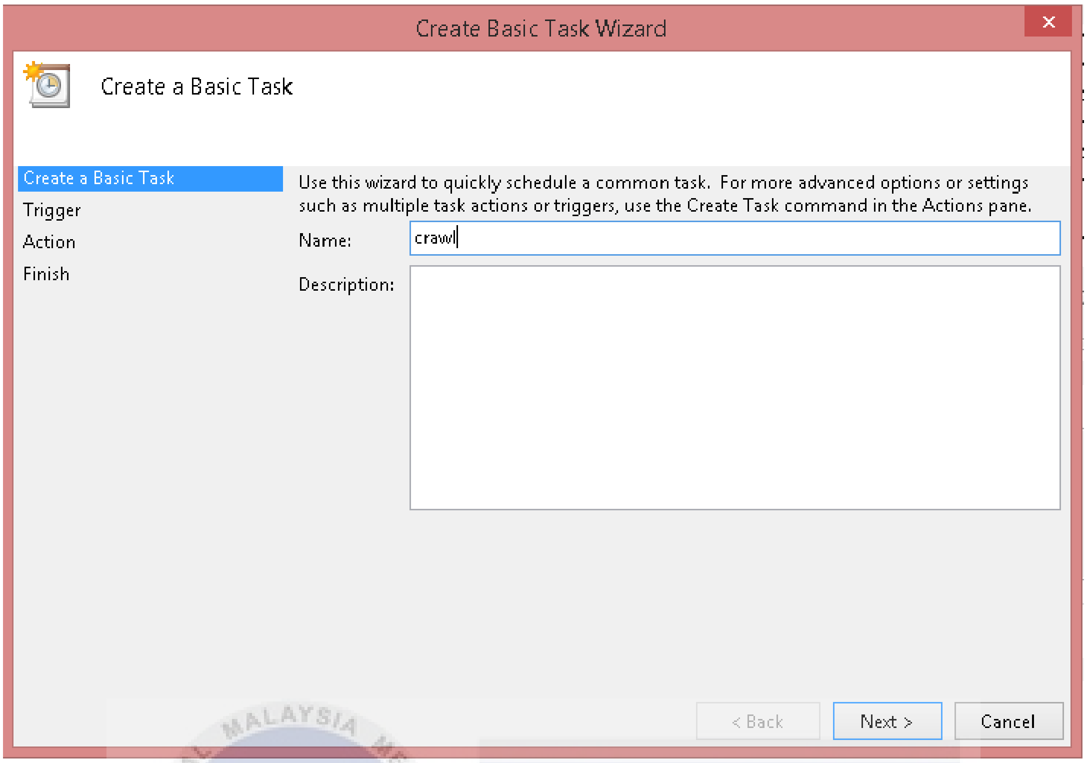
Test database connection.



```
C:\Python27\python.exe "C:/Users/nurul fadhilah/PycharmProjects/myproject/connect.py"
You have the connection to database!
Database version : 10.1.21-MariaDB

Process finished with exit code 0
```

Figure 5.5: Output of database connection

Next phase, we need to create a scheduler for crawler.



Create Basic Task Wizard

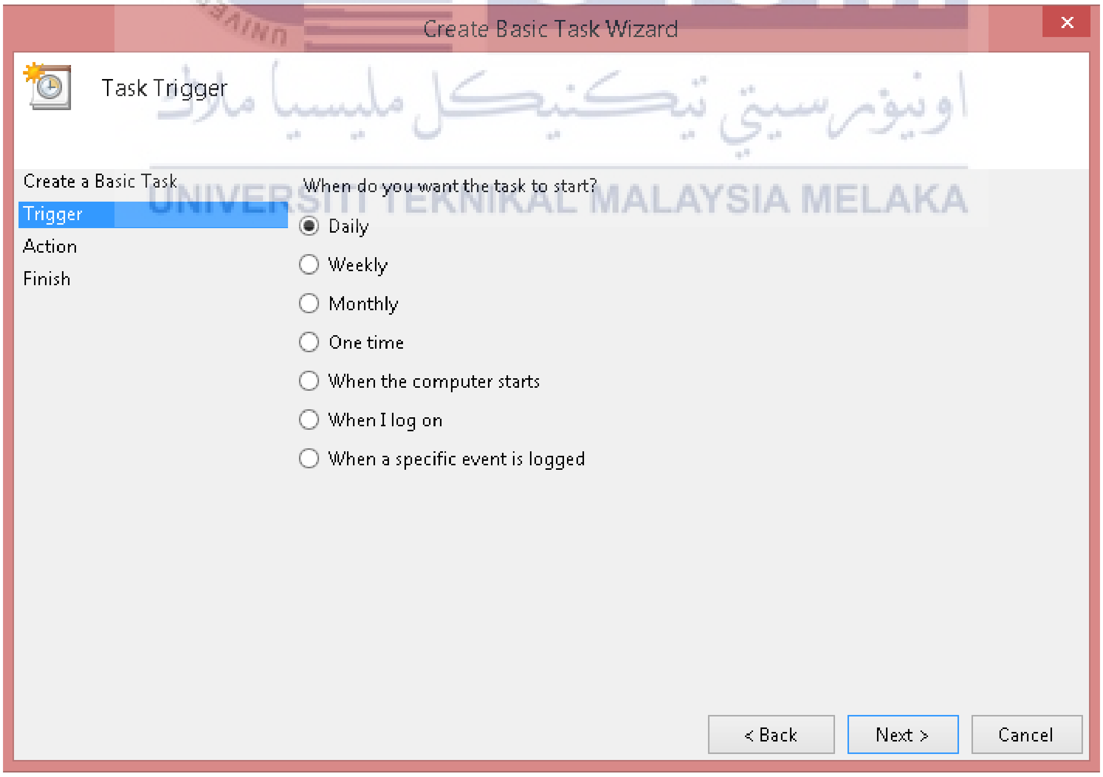
Create a Basic Task

Use this wizard to quickly schedule a common task. For more advanced options or settings such as multiple task actions or triggers, use the Create Task command in the Actions pane.

Name: crawl

Description:

Figure 5.6: Create task name



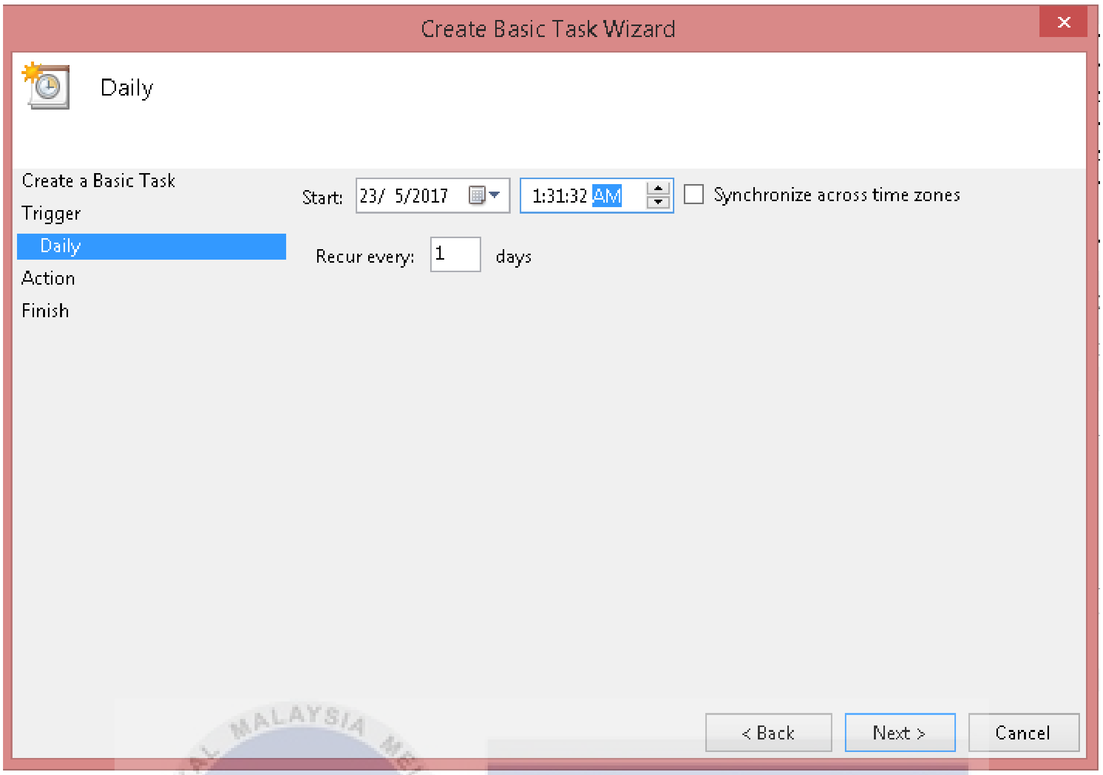
Create Basic Task Wizard

Task Trigger

When do you want the task to start?

● Daily
○ Weekly
○ Monthly
○ One time
○ When the computer starts
○ When I log on
○ When a specific event is logged

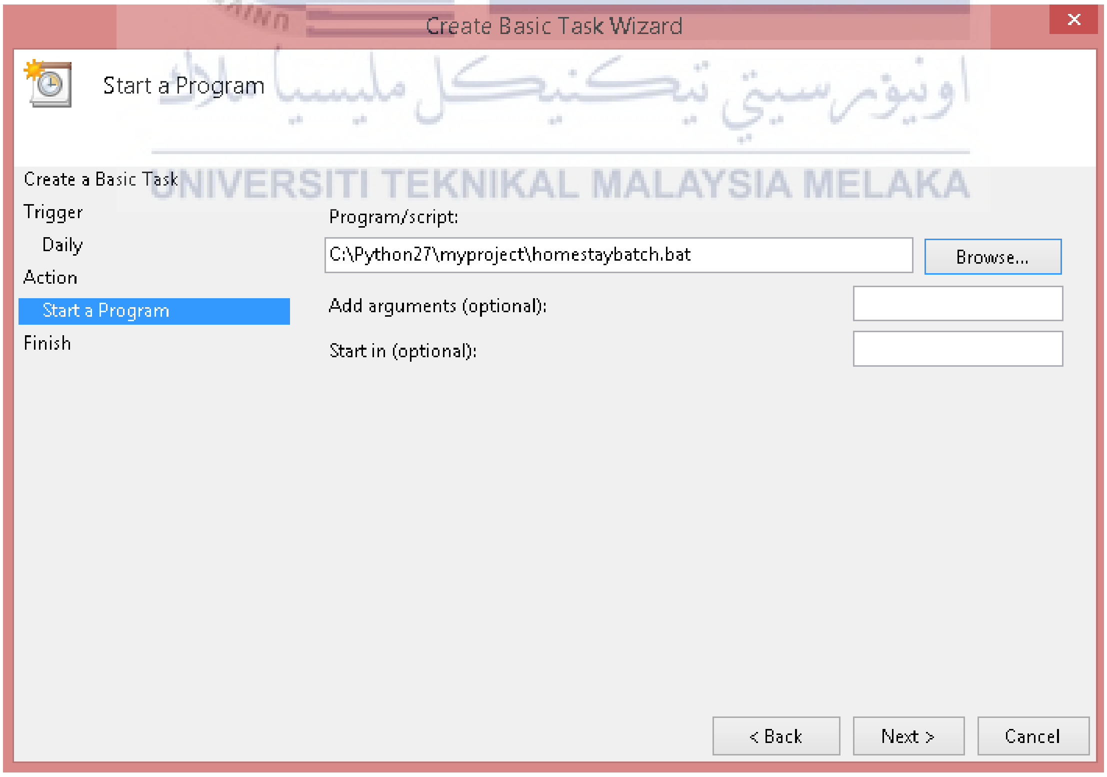
Figure 5.7: Scheduler set up



Start: 23/ 5/2017 1:31:32 AM □ Synchronize across time zones

Recur every: 1 days

Figure 5.8: Scheduler time set up



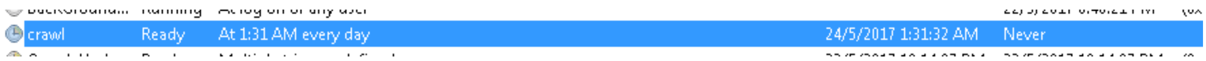
Program/script: C:\Python27\myproject\homestaybatch.bat

Add arguments (optional):

Start in (optional):

Figure 5.9: Scheduler script

A scheduler that run the crawler program has been successfully created.



```
crawl     Ready     At 1:31 AM every day     24/5/2017 1:31:32 AM     Never
```

Figure 5.10: Complete Scheduler

5.3.2 Version Control Procedure

Table 5.1 Version control of homestay crawler

Version	Description
Version 1.0	Provide simple functions to retrieve title and price of homestay from web marketplace.
Version 1.1	Added features to retrieve state and district of homestay.
Version 1.2	All information needed can be retrieved.

Table 5.2 Version control of homestay statistics dashboard

Version	Description
Version 1.0	Integration of data retrieved by crawler from web marketplace with displaying chart comparison.
Version 1.1	Addition of tables displaying information of homestay for all states in Malaysia.
Version 1.2	More charts and tables added for more detailed data comparison.

5.4 Implementation Status

Table 5.3 Implementation status of system

Implementation	**Description**	**Duration(week)**	**Date completed**
Development of web crawler	Implement web scraping method.	4	6/3/2017
Data collection and extraction	Collect and gather information needed.	3	28/4/2017
Repository development	Develop a database to store all information retrieved.	2	5/5/2017
Platform to visualize data	Develop a platform that act as a reporting tool.	5	18/7/2017
Implementation and testing	Validates the information crawled to be accurate.	4	9/8/2017

5.5 Conclusion

Overall, the implementation status of homestay crawler and statistics dashboard are within milestone. All the system and methods have been implemented successfully. In the next chapter will be discussed on testing of the system.

CHAPTER VI

TESTING

6.1 Introduction

In previous chapter, we have discussed briefly about the implementation of the system. Now, we will proceed to discuss the testing phase of the project. This phase is significant to project to ensure that the system is completed, smooth and meet the requirements of project.

6.2 Test Plan

In this section, the basis of the system is explained. Test case, testing scope and activities is carried out for this phase.

6.2.1 Test Organization

Personnel involved in testing of the system crawler and dashboard according to specific requirements.

Table 6.1 Test organization

Personnel	**Role**	**Description**
Nurul Fadhilah Sainan	System developer	Develop the system and involved in system testing of homestay crawler.
Dr. Zurina Sa'aya	Supervisor	Supervise and involved in system testing.

6.2.2 Test Environment

The test environment of crawler and dashboard is set up according to system architecture discussed in previous chapter to perform the testing of crawler and dashboard that have been developed.

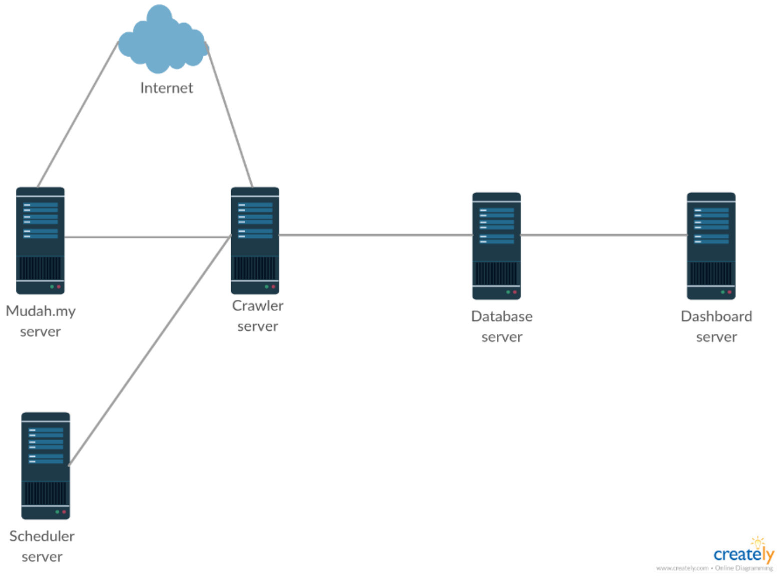


Figure 6.1: Structure of test environment

Table 6.2: Description of test environment

Item	Description
Mudah.my Server	Server that kept all information regarding homestay.
Homestay Crawler Server	Server that function as a platform to run and test the homestay crawler that extract information from the web marketplace.
Database Server	Local repository that store all retrieved information from crawler.

6.2.3 Test Schedule

Table 6.3: Test Schedule

No	Item	Duration	Dates	Person in-charge
1	Team setup and kick start	1 day	1/8/2017	Nurul Fadhilah
2	Software Test Documentation(STD)		3-4/8/2017	Nurul Fadhilah

No	Item	Duration	Dates	Person in-charge
3	Test cases design and development	2 days	6-7/8/2017	Nurul Fadhilah
4	Test data preparation	2 days	8-9/8/2017	Nurul Fadhilah
5	Environment setup	3 days	11-13/8/2017	Nurul Fadhilah
6	Testing deployment	2 days	14-15/8/2017	Nurul Fadhilah
7	Software Test Result(STR)	2 days	16-17/8/2017	Nurul Fadhilah

6.3 Test Strategy

The black-box testing procedure is used to test the usefulness of the system framework disregarding the inside structure of the framework. It is appropriate to use for this case since black-box testing bargains practically every high level testing. It should be possible by giving the system a data and analyze the exactness of the output.

6.4 Test Design

This topic explain the testing design for the black-box testing strategy of homestay crawler and analytics dashboard. Homestay information of the first page of website are crawled and tested by different condition.

6.4.1 Test Description

The test case is carried out for the crawl function. This testing phase will verify system functionality.

Table 6.4: Description of test case of crawler

Features	**Test steps**	**Expected result**
Crawling	Run main.py in PyCharm software with the data mentioned above.	All available information will be stored into local repository system.

The source code can be viewed at Appendices B

Table 6.5: Description of test case of statistics dashboard

Features	**Test steps**	**Expected result**
Dashboard	Open URL of the index page of homestay analytics dashboard using a web browser.	Data retrieved by homestay crawler will be visualized into website.
Data tables	Open the URL of country.php page of homestay analytics dashboard using a web browser.	Data retrieved by homestay crawler will be visualized in form of tables.
Morris.js chart	Open the URL of (state name).php page of homestay analytics dashboard using a web browser.	Data retrieved by homestay crawler will be visualized in form of pie charts.

The result can be viewed at Appendices A

The source code can be viewed at Appendices B

The test case can be viewed at Appendices D

Table 6.6: Description of test case of scheduler

Features	Test steps	Expected result
Task scheduler	Task scheduler is set to run a program at specified time.	A task will appear at windows task scheduler and error message displayed if failed. The updater will run every once daily.

Table 6.7: Description of test case of server connection

Features	Test steps
Dashboard server	Open URL in web browser with/without internet connection

The result can be viewed at Appendices A

The test case can be viewed at Appendices D

Table 6.8: Description of test case of server connection with different browser

Scenario	Google chrome	Internet explorer	Mozilla Firefox
Dashboard server	Compatible	Compatible	Compatible

The result can be viewed at Appendices A

The test case can be viewed at Appendices D

6.4.2 Test Data

The test data used is homestay information collected from online public marketplace Mudah.my which consists of homestay information of the first page of website are crawled and tested by different condition.

6.5 Test Results and Analysis

The result can be viewed at Appendices A

The test case can be viewed at Appendices D

6.6 Conclusion

Testing phase is to ensure that errors are identified and corrected if any. All server can function if there are internet connection. Browser testing also show positive results which is compatible to run servers used in this project.

CHAPTER VII

PROJECT CONCLUSION

7.1 Introduction

This chapter summarizes the project, explanation about how objectives been achieved from implementation and testing. Discussion of weakness and strength of project, project contribution to company or individual. Explanation of project limitation and future project improvement.

7.2 Project Summarization

Both homestay crawler and statistics dashboard developed has satisfied the project objectives. The first objective which is to develop a web crawler to retrieve information of homestay from online public marketplace which is Mudah.my is achieved using homestay crawler which crawl information in timely manner.

Second objective which is to extract information from the retrieved data and build a local repository to store the information is achieved by using MySQL database. All information retrieved is inserted in the database.

Third objective which is to develop a platform to summarize and visualize homestay information also achieved. All information retrieved is visualized in chart form to provide more detailed and specific data comparison.

There are several weakness in the system. It takes a very long process to crawl very big amount of data at a time. Also, if the structure of the online public marketplace changes, this crawler might be unusable at that time.

The strength of the system is that the homestay information updated daily so we get a collection of updated information from Mudah.my. Although there might be some data redundancy.

7.3 Project Contribution

At the end, this project develop a set of graphical presentation of homestays advertisement information. With the availability of this information, user can better monitor and stay updated with homestays information in just one platform and plan towards increasing the number of tourists visiting Malaysia that will stay at homestays during their visit and ultimately improve Malaysia tourism ranking. Therefore, this dashboard is a powerful tool for tourism department and city council to assist them for decision making and promoting.

7.4 Project Limitation

For the current system that has been developed, the homestay crawler will only scrap data from the first page of information from the website. Supposedly, the system will crawl all pages of the website with short process but because of time limitation the crawler is set to scrap only one page. This system can crawl to all pages but the process will take time since the crawler need to loop according to number of pages.

7.5 Future Works

More features can be included into the homestay analytics dashboard to display more detailed and specific data comparison. Also, diversifying data presentation by using different charts. More integration between crawler and dashboard can be added to enhance the system functionality. Also, adding system function such as date of advertisement and detailed redundancy checking.

7.6 Conclusion

To conclude, all the project objectives has been achieved successfully. With the help of homestay crawler and statistics dashboard, user can better monitor homestay of each states in Malaysia advertised in Mudah.my in a more practical way.

REFERENCES

[1] Edwards, J., Mccurley, K., & Tomlin, J. (2001). An Adaptive Model for Optimizing Performance of an Incremental Web Crawler. *WWW '01 Proceedings of the 10th International Conference on World Wide Web*, 106–113. https://doi.org/10.1145/371920.371960

[2] Mahar, B., & Jha, C. K. (2015). A Comparative Study on Web Crawling for searching Hidden Web, *6*(3), 2159–2163.

[3] Verma, A. (2016). Breadth-first search crawling yields high-quality pages, (October). Retrieved from http://www10.org/cdrom/papers/frame.html

[4] Bangi, U. K. M., & Ehsan, D. (2011). The Role of Home Stay for Ecotourism Development in East Coast Economic Region Md . Anowar Hossain Bhuiyan , Chamhuri Siwar , Shaharuddin Mohamad Ismail and Rabiul Islam Institute for Environment and Development , University Kebangsaan Malaysia , *8*(6), 540–546.

[5] Boldi, P., Codenotti, B., Santini, M., & Vigna, S. (2003). UbiCrawler□: a scalable fully distributed web crawler. *Software: Practice & Experience*, 1–18. https://doi.org/10.1002/spe.587

[6] Shkapenyuk, V., & Suel, T. (2002). Design and implementation of a high-performance distributed Web crawler. *Icde 2002*, 357–368. https://doi.org/10.1109/ICDE.2002.994750

[7] Sun, Y., Councill, I.G. & Giles, C.L., 2010. The ethicality of web crawlers. Proceedings - 2010 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010, 1, pp.668–675.

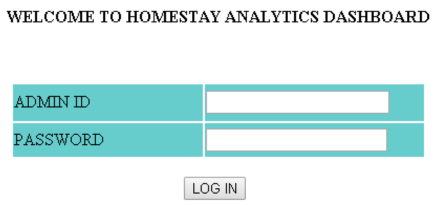
[8] Kenneth Reitz (2016) HTML scraping — the Hitchhiker's guide to python. Available at: http://docs.python-guide.org/en/latest/scenarios/scrape/

[9] Richardson, L. (2015) Beautiful soup: We called him tortoise because he taught us. Available at: https://www.crummy.com/software/BeautifulSoup/

[10] YJ YU, (2012), Homestay Online Booking and Management System. Available at: http://umpir.ump.edu.my/5101/1/CD6548.pdf

APPENDICES A: USER MANUAL

Login interface



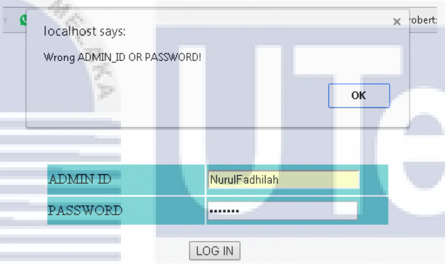
WELCOME TO HOMESTAY ANALYTICS DASHBOARD

ADMIN ID

PASSWORD

Figure shows that admin need to enter id and password in order to have access to the system.

Invalid password



localhost says:

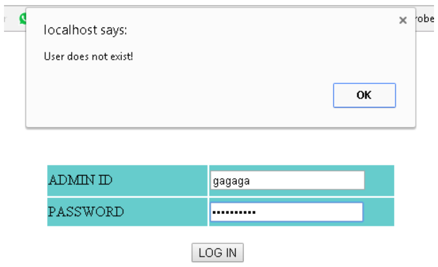
Wrong ADMIN_ID OR PASSWORD!

ADMIN ID: NurulFadhilah

PASSWORD: •••••••

Figure shows that if user enter the wrong password, user unable to have access to the system. A pop-up of error message that says the password is wrong will show up.

Invalid admin id



localhost says:

User does not exist!

ADMIN ID: gagaga

PASSWORD: ••••••••••

Figure shows that if user enter the wrong user id, user unable to have access to the system. A pop-up of error message that says the user does not exist will show up.

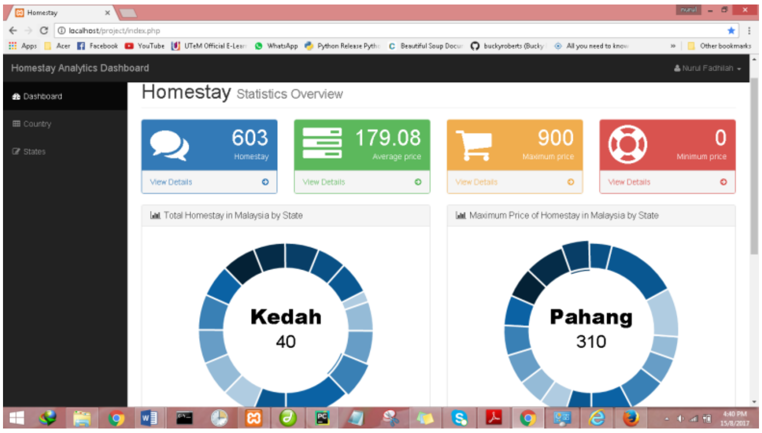

Homestay Analytics Dashboard
Dashboard
Country
States
Homestay Statistics Overview
603 Homestay
179.08 Average price
900 Maximum price
0 Minimum price
View Details
Total Homestay in Malaysia by State
Maximum Price of Homestay in Malaysia by State
Kedah 40
Pahang 310

Figure shows dashboard server open with internet connection using Google Chrome browser

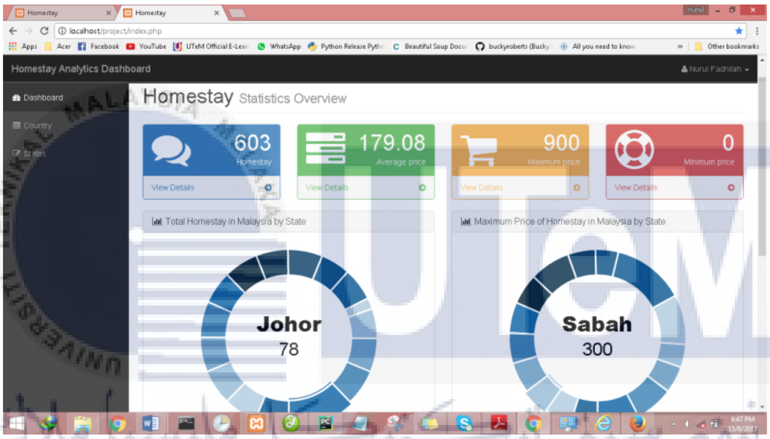

Homestay Analytics Dashboard
Dashboard
Country
States
Homestay Statistics Overview
603 Homestay
179.08 Average price
900 Maximum price
0 Minimum price
View Details
Total Homestay in Malaysia by State
Maximum Price of Homestay in Malaysia by State
Johor 78
Sabah 300

Figure shows dashboard server open without internet connection using Google Chrome browser

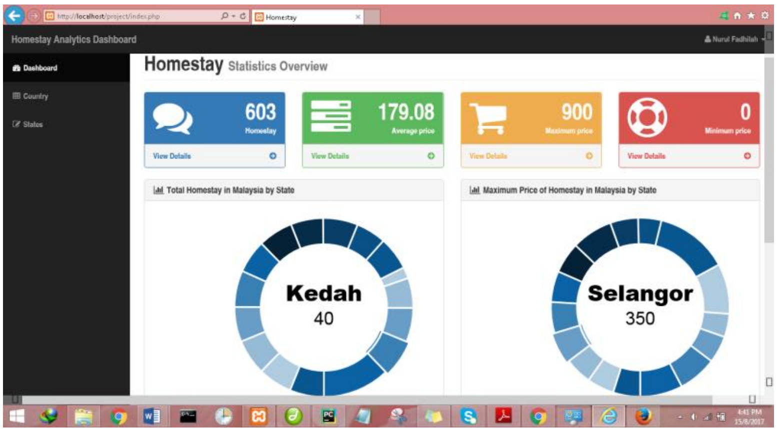

Homestay Analytics Dashboard
Dashboard
Country
States
Homestay Statistics Overview
603 Homestay
179.08 Average price
900 Maximum price
0 Minimum price
View Details
Total Homestay in Malaysia by State
Maximum Price of Homestay in Malaysia by State
Kedah 40
Selangor 350

Figure shows dashboard server open with internet connection using Internet Explorer browser

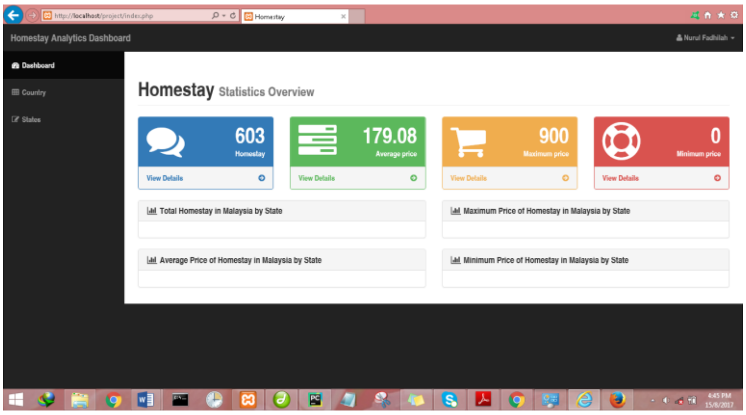

http://localhost/project/index.php
Homestay
Homestay Analytics Dashboard
Nurul Fadhilah
Dashboard
Country
States
Homestay Statistics Overview
603
Homestay
179.08
Average price
900
Maximum price
0
Minimum price
View Details
Total Homestay in Malaysia by State
Maximum Price of Homestay in Malaysia by State
Average Price of Homestay in Malaysia by State
Minimum Price of Homestay in Malaysia by State

Figure shows dashboard server open without internet connection using Internet Explorer browser

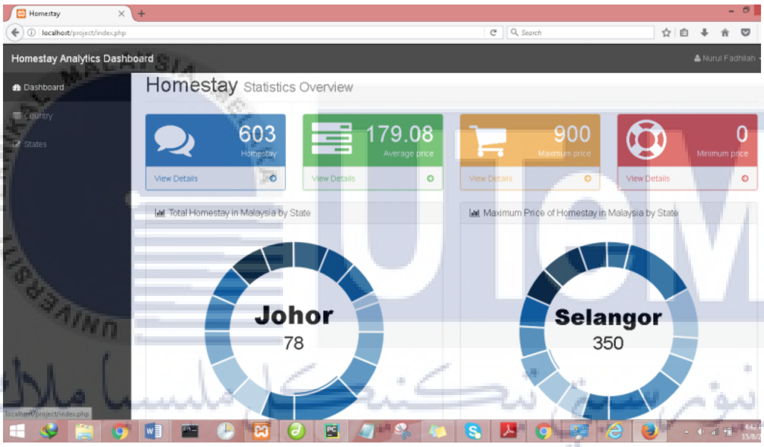

Homestay
localhost/project/index.php
Search
Homestay Analytics Dashboard
Nurul Fadhilah
Dashboard
Country
States
Homestay Statistics Overview
603
Homestay
179.08
Average price
900
Maximum price
0
Minimum price
View Details
Total Homestay in Malaysia by State
Maximum Price of Homestay in Malaysia by State
Johor
78
Selangor
350

Figure shows dashboard server open with internet connection using Mozilla Firefox browser

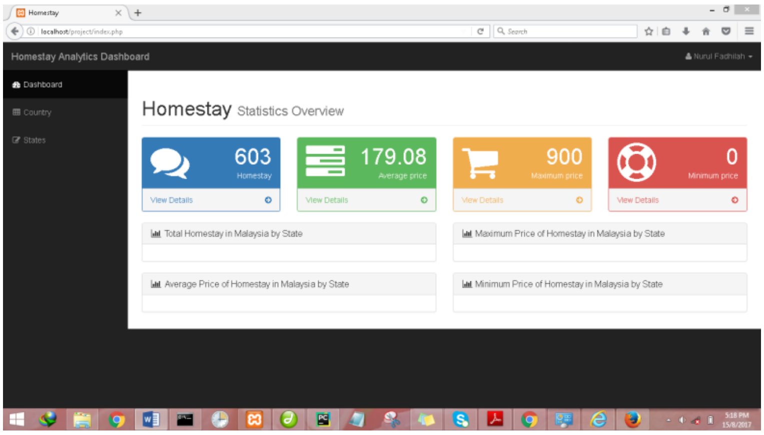
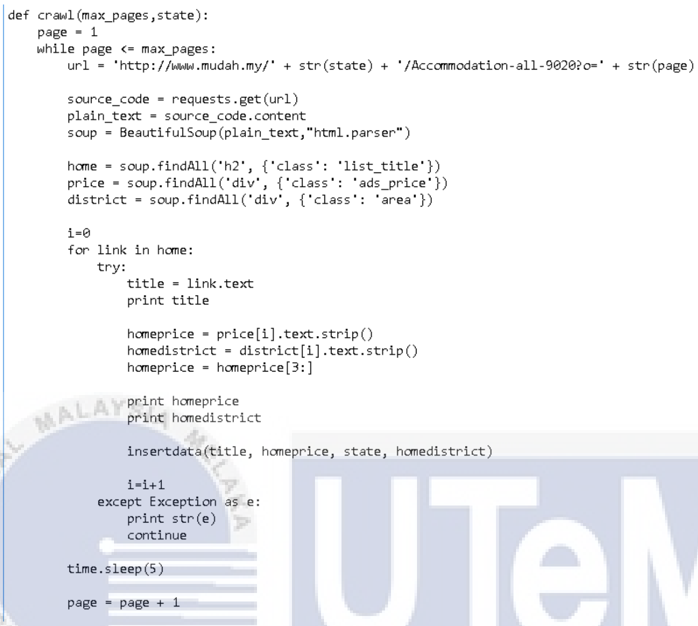

Homestay
localhost/project/index.php
Search
Homestay Analytics Dashboard
Nurul Fadhilah
Dashboard
Country
States
Homestay Statistics Overview
603
Homestay
179.08
Average price
900
Maximum price
0
Minimum price
View Details
Total Homestay in Malaysia by State
Maximum Price of Homestay in Malaysia by State
Average Price of Homestay in Malaysia by State
Minimum Price of Homestay in Malaysia by State

Figure shows dashboard server open without internet connection using Mozilla Firefox browser

APPENDICES B: SOURCE CODE

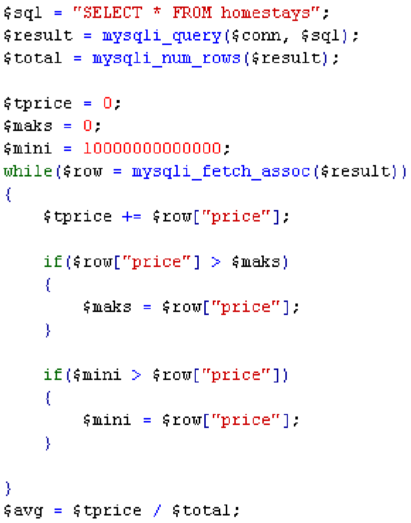
Homestay crawler



```
def crawl(max_pages,state):
    page = 1
    while page <= max_pages:
        url = 'http://www.mudah.my/' + str(state) + '/Accommodation-all-9020?o=' + str(page)

        source_code = requests.get(url)
        plain_text = source_code.content
        soup = BeautifulSoup(plain_text,"html.parser")

        home = soup.findAll('h2', {'class': 'list_title'})
        price = soup.findAll('div', {'class': 'ads_price'})
        district = soup.findAll('div', {'class': 'area'})

        i=0
        for link in home:
            try:
                title = link.text
                print title

                homeprice = price[i].text.strip()
                homedistrict = district[i].text.strip()
                homeprice = homeprice[3:]

                print homeprice
                print homedistrict

                insertdata(title, homeprice, state, homedistrict)

                i=i+1
            except Exception as e:
                print str(e)
                continue

        time.sleep(5)

        page = page + 1
```

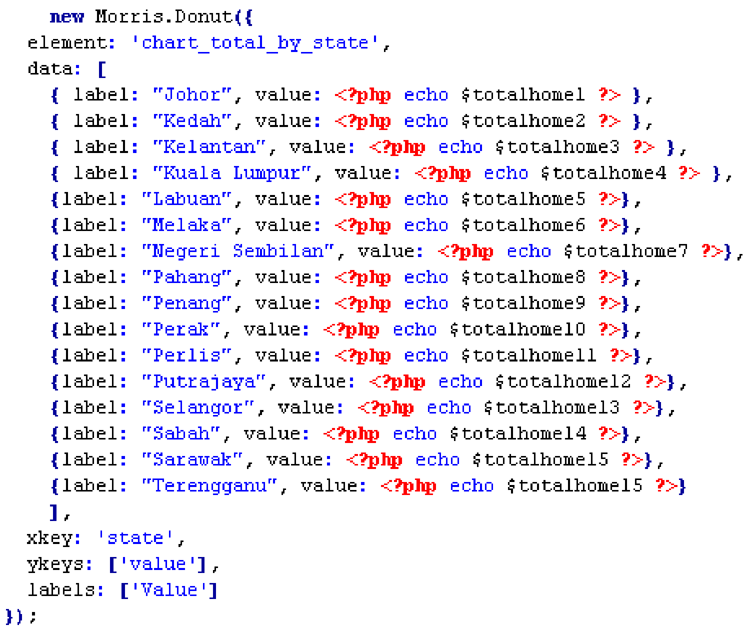
Figure shows the python source code function to crawl information from website.

Dashboard



```
$sql = "SELECT * FROM homestays";
$result = mysqli_query($conn, $sql);
$total = mysqli_num_rows($result);

$tprice = 0;
$maks = 0;
$mini = 10000000000000;
while($row = mysqli_fetch_assoc($result))
{
    $tprice += $row["price"];

    if($row["price"] > $maks)
    {
        $maks = $row["price"];
    }

    if($mini > $row["price"])
    {
        $mini = $row["price"];
    }

}
$avg = $tprice / $total;
```

Figure shows the SQL function to calculate total homestay, average, maximum and minimum price.

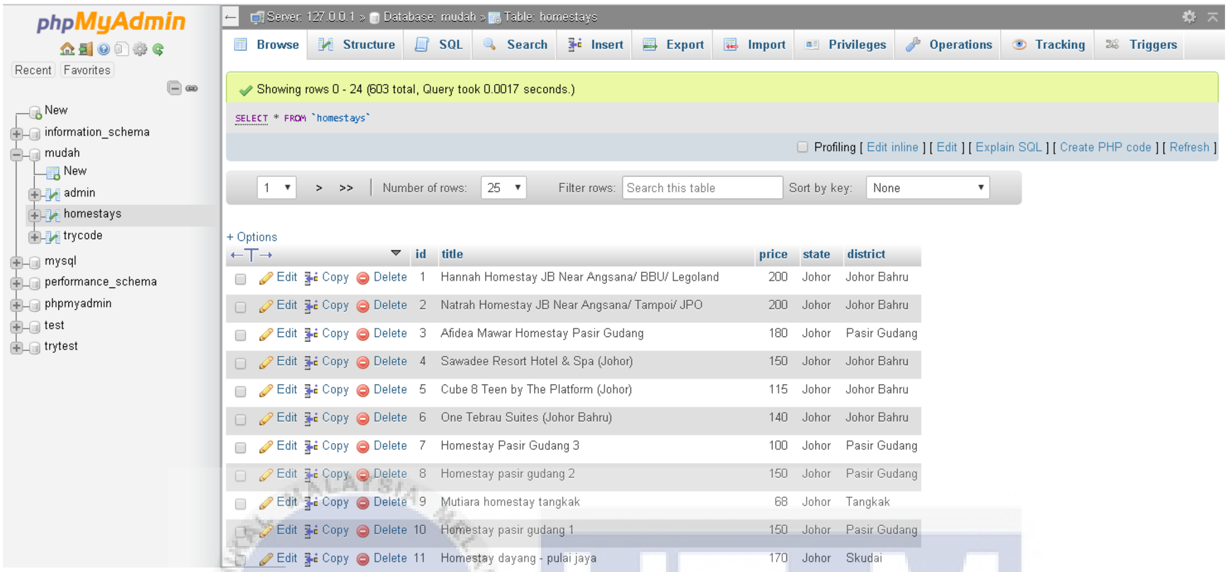


```
     new Morris.Donut({
   element: 'chart_total_by_state',
   data: [
     { label: "Johor", value: <?php echo $totalhome1 ?> },
     { label: "Kedah", value: <?php echo $totalhome2 ?> },
     { label: "Kelantan", value: <?php echo $totalhome3 ?> },
     { label: "Kuala Lumpur", value: <?php echo $totalhome4 ?> },
     {label: "Labuan", value: <?php echo $totalhome5 ?>},
     {label: "Melaka", value: <?php echo $totalhome6 ?>},
     {label: "Negeri Sembilan", value: <?php echo $totalhome7 ?>},
     {label: "Pahang", value: <?php echo $totalhome8 ?>},
     {label: "Penang", value: <?php echo $totalhome9 ?>},
     {label: "Perak", value: <?php echo $totalhome10 ?>},
     {label: "Perlis", value: <?php echo $totalhome11 ?>},
     {label: "Putrajaya", value: <?php echo $totalhome12 ?>},
     {label: "Selangor", value: <?php echo $totalhome13 ?>},
     {label: "Sabah", value: <?php echo $totalhome14 ?>},
     {label: "Sarawak", value: <?php echo $totalhome15 ?>},
     {label: "Terengganu", value: <?php echo $totalhome15 ?>}
     ],
   xkey: 'state',
   ykeys: ['value'],
   labels: ['Value']
 });
```

Figure shows the script to insert pie chart into dashboard.

APPENDICES C: DATABASE

Homestay database



✓ Showing rows 0 - 24 (603 total, Query took 0.0017 seconds.)

```
SELECT * FROM `homestays`
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

		id	title	price	state	district
□	Edit Copy Delete	1	Hannah Homestay JB Near Angsana/ BBU/ Legoland	200	Johor	Johor Bahru
□	Edit Copy Delete	2	Natrah Homestay JB Near Angsana/ Tampoi/ JPO	200	Johor	Johor Bahru
□	Edit Copy Delete	3	Afidea Mawar Homestay Pasir Gudang	180	Johor	Pasir Gudang
□	Edit Copy Delete	4	Sawadee Resort Hotel & Spa (Johor)	150	Johor	Johor Bahru
□	Edit Copy Delete	5	Cube 8 Teen by The Platform (Johor)	115	Johor	Johor Bahru
□	Edit Copy Delete	6	One Tebrau Suites (Johor Bahru)	140	Johor	Johor Bahru
□	Edit Copy Delete	7	Homestay Pasir Gudang 3	100	Johor	Pasir Gudang
□	Edit Copy Delete	8	Homestay pasir gudang 2	150	Johor	Pasir Gudang
□	Edit Copy Delete	9	Mutiara homestay tangkak	68	Johor	Tangkak
□	Edit Copy Delete	10	Homestay pasir gudang 1	150	Johor	Pasir Gudang
□	Edit Copy Delete	11	Homestay dayang - pulai jaya	170	Johor	Skudai

APPENDICES D: TEST CASE

Module Name: Login

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS
LOGIN_01	Valid id and password	Enter valid admin id and valid password.		1.Enter id 2.Enter password 3.Click “LOG IN” button	<valid id> <valid password>	Successful login		Successful login	PASS
LOGIN_02	Valid id and password	Enter valid admin id and invalid password.		1.Enter id 2.Enter password 3.Click “LOG IN” button	<valid id> <invalid password>	A message "Wrong PASSWORD!" is shown		A message "Wrong PASSWORD!" is shown	PASS

LOGIN _03	Valid id and password	Enter invalid admin id and valid password.		1.Enter id 2.Enter password 3.Click "LOG IN" button	<invalid id> <valid password>	A message "User does not exist!" is shown		A message "User does not exist!" is shown	PASS
LOGIN _04	Valid id and password	Enter invalid admin id and invalid password.		1.Enter id 2.Enter password 3.Click "LOG IN" button	<invalid id> <invalid password>	A message "Wrong ADMIN_ID OR PASSWORD!" is shown		A message "Wrong ADMIN_ID OR PASSWORD!" is shown	PASS

Name: Google Chrome browser

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS
DASHBOARD_01	Strong internet connection	Open dashboard index page with internet connection.		1.open browser. 2.enter URL	Strong internet connection	Access to server successfully.		Access to server successfully.	PASS
DASHBOARD_02	Strong internet connection	Open dashboard index page without internet connection.		1.open browser. 2.enter URL	Strong internet connection	Access to server successfully.		Access to server successfully.	PASS

Module Name: Internet Explorer browser

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS
IE_01	Strong internet connection	Open dashboard index page with internet connection.		1.open browser. 2.enter URL	Strong internet connection	Access to server successfully.		Access to server successfully.	PASS
IE_02	Strong internet connection	Open dashboard index page without internet connection.		1.open browser. 2.enter URL	Strong internet connection			Cannot access to server successfully.	PASS

Module Name: Mozilla Firefox browser

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS
FIREFOX_01	Strong internet connection	Open dashboard index page with internet connection.		1.open browser. 2.enter URL	Strong internet connection	Access to server successfully.		Access to server successfully.	PASS
FIREFOX_02	Strong internet connection	Open dashboard index page without internet connection.		1.open browser. 2.enter URL	Strong internet connection	Cannot access to server successfully		Cannot access to server successfully.	PASS