



# GROUP ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

AAPP006-4-2-SDP

SOFTWARE DEVELOPMENT PROJECT

## PART 2

**HAND OUT DATE: 10 JANUARY 2022**

**HAND IN DATE: 08 APRIL 2022**

**WEIGHTAGE: 100%**

---

### INSTRUCTIONS TO CANDIDATES:

1. Submit your assignment at the administrative counter.
2. Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing).
3. Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.
4. Cases of plagiarism will be penalized.
5. The assignment should be bound in an appropriate style (comb bound or stapled).
6. Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.
7. You must obtain 50% overall to pass this module.



## Software Development Project – Cover Sheet ASSIGNMENT

**Instruction:**

- Marks will be awarded for good presentation and thoroughness in your approach.
- Referencing Code: If you use some code, or ideas for code, which are taken or adapted from another source (book, magazine, internet, discussion forum, etc.), then this **must** be cited and referenced using the APA convention within your source code. Failure to reference code properly is considered as plagiarism.
- Complete this cover sheet and attach it to your project first page.
- This project is to be attempted by a group of 4 - 5 students.

**Student declaration:**

<i>I declare that:</i>	<ul style="list-style-type: none"> <li>▪ We understand what is meant by plagiarism</li> <li>▪ The implication of plagiarism have been explained to us by our lecturer</li> <li>▪ This project is all our work and we have acknowledged any use of the published or unpublished works of other people.</li> </ul>
Group Leader's Signature: <i>Teo Kai Yii</i> .....	Date: <i>5/4/2022</i> .....

<b>Project Title:</b> Valley Clinic Management System		<b>Intake:</b> UCDF2005(1)ICT(SE)
Role	Name	Signature
Project Manager	TEO KAI YII	<i>Teo Kai Yii</i>
System Analyst	TAN KAR WEI	<i>karwei</i>
Programmer	HOOI KAI JUN	<i>Hooi Kai Jun</i>
Programmer	SAM ZHENG YEW	<i>Sam Zheng Yew</i>
System Tester	NISHARINI A/P GANESAN	<i>Nisharini Ganesan</i>

## **2.0 Table of Contents**

1.0 Cover Page .....	2
2.0 Table of Contents.....	3
3.0 Acknowledgement .....	11
4.0 Abstract .....	12
5.0 Workload Matrix.....	13
6.0 Introduction.....	14
6.1 Project Background.....	14
6.2 Problem Context .....	15
6.3 Proposed Solution .....	17
6.4 Project Aim and Objectives .....	19
6.5 Project Scope .....	20
6.6 Product Scope .....	21
7.0 Project Plan .....	22
7.1 System Development Methodology.....	22
7.2 Project Gantt Chart .....	25
8.0 System Hierarchy Chart.....	26
9.0 Context Diagram.....	27
10.0 Data Flow Diagrams Level 0 .....	28
11.0 Data Flow Diagrams Level 1 .....	29
11.1 Teo Kai Yii - TP058618 .....	29
11.2 Sam Zheng Yew - TP060764.....	30
11.3 Hooi Kai Jun – TP060766.....	31
11.4 Tan Kar Wei – TP060354.....	32
11.5 Nisharini Ganesan – TP059071 .....	33

12.0 Data Dictionary .....	34
12.1 Teo Kai Yii - TP058618 .....	34
12.2 Sam Zheng Yew - TP060764.....	38
12.3 Hooi Kai Jun - TP060766 .....	42
12.4 Tan Kar Wei – TP060354.....	45
12.5 Nisharini Ganesan – TP059071 .....	48
13.0 Entity Relationship Diagram.....	51
14.0 Pseudocode or Flowcharts .....	53
14.1 Teo Kai Yii – TP058618.....	53
14.2 Sam Zheng Yew - TP060764.....	54
14.3 Hooi Kai Jun – TP060766.....	56
14.4 Tan Kar Wei – TP060354.....	57
14.5 Nisharini Ganesan - TP059071 .....	58
15.0 Screen Design and User Manual.....	59
15.1 Customer Module.....	59
15.1.1 Customer Booking .....	59
15.1.2 Customer Confirm Booking Details .....	60
15.1.3 Payment Gateway .....	61
15.1.4 Customer Homepage.....	62
15.1.5 About Us .....	64
15.1.6 Contact Us.....	66
15.1.7 Treatment And Services.....	68
15.1.8 Medical Condition .....	70
15.1.9 Our Doctor .....	72
15.1.10 Sign In.....	74

15.1.11 Sign Up .....	75
15.1.12 Redirect User Index .....	76
15.1.13 User Profile .....	77
15.1.15 Booking History.....	81
15.1.15 Booking Invoice.....	82
15.2 Administrator Module.....	83
15.2.1 Login.....	83
15.2.2 Dashboard .....	84
15.2.3 Manage Treatment & Services.....	85
15.2.4 Manage Administrator .....	89
15.2.5 Manage Staff.....	91
15.2.6 Manage Doctor.....	93
15.2.7 Reports .....	96
15.2.8 Logout.....	98
15.3 Doctor Module .....	99
15.3.1 Doctor Login.....	99
15.3.2 Doctor Homepage .....	100
15.3.2 Doctor Appointment Page.....	101
15.3.3 Doctor Treatment Record Page.....	102
15.3.4 Doctor Account Page .....	106
15.3.5 Doctor Logout Page .....	108
15.4 Staff Module .....	109
15.4.1 Staff Login .....	109
15.4.2 Staff Dashboard Overview.....	110
15.4.3 Add Medicine.....	111

15.4.4 Search Medicine.....	112
15.4.5 Edit Medicine.....	113
15.4.6 Delete Medicine .....	114
15.4.7 Check Appointment .....	115
15.4.8 Staff Logout .....	116
16.0 Report Designs and User Manual .....	117
16.1 Customer Module.....	117
16.1.1 Navigation Bar.....	117
16.1.2 Footer .....	117
16.1.3 Customer Homepage.....	118
16.1.4 About Us .....	119
16.1.5 Contact Us.....	121
16.1.6 Treatment and Services.....	123
16.1.7 Medical Services detail .....	125
16.1.8 Our Doctors.....	126
16.1.9 User Notification on requirement (Login/Sign Up).....	127
16.1.10 User Sign In .....	127
16.1.11 User Sign-Up .....	129
16.1.11 Booking and Appointment.....	130
16.1.12 Confirm Appointment Details.....	132
16.1.13 Payment Gateway .....	133
16.2 Administrator Module.....	136
16.2.1 Login .....	136
15.2.2 Dashboard .....	137
15.2.3 Treatment & Services .....	138

15.2.4 Manage Administrator .....	145
15.2.5 Manage Staff.....	149
15.2.6 Manage Doctor.....	153
15.2.7 Reports .....	158
15.2.8 Logout.....	159
16.3 Doctor Module .....	161
16.3.1 Doctor's Login .....	161
16.3.2 Doctor's Homepage .....	162
16.3.3 Side Navigation Bar .....	164
16.3.3 Doctor's Appointment Page.....	164
16.3.4 Doctor's Treatment Record Page.....	165
16.3.5 Doctor's Account Page .....	168
16.3.5 Logout.....	171
16.4 Staff Module .....	172
16.4.1 Staff Login .....	172
16.4.2 Staff Overview .....	173
16.4.3 Add Medicine.....	174
16.4.4 Edit Medicine.....	175
16.4.5 Search Medicine.....	176
16.4.6 Delete Medicine .....	177
16.4.7 Check Appointment .....	178
16.4.8 Logout.....	179
17.0 Test Plan (Unit Testing).....	180
17.1 Customer Module.....	180
17.2 Administrator Module.....	184

17.3 Doctor Module .....	191
17.4 Staff Module .....	198
18.0 Significant Source Code .....	204
18.1 Bootstrap .....	204
18.2 Bootstrap Icon .....	206
18.3 Font Awesome Icon .....	207
18.4 Chart.js .....	209
18.5 DataTables .....	210
18.6 Customer Module.....	211
18.6.1 Create – HTML Form, PHP.....	211
18.6.2 Login – HTML, Form, PHP.....	217
18.6.3 Signup – HTML, Form, PHP .....	220
18.6.4 Update – HTML, Form, PHP.....	223
18.6.5 Read – HTML, Form, PHP .....	227
18.6.6 Self-created CSS and JavaScript.....	234
18.7 Administrator Module.....	258
18.7.1 Create – HTML Form, PHP.....	258
18.7.2 Read – HTML Form, PHP .....	266
18.7.3 Update – HTML Form, PHP .....	281
18.7.4 Delete – HTML Form, PHP.....	284
18.7.5 Login – HTML Form, PHP.....	285
18.7.6 Signup – HTML Form, PHP .....	289
18.7.8 Self-created CSS and JavaScript.....	291
18.8 Doctor Module .....	296
18.8.1 Create – HTML Form, PHP.....	296

18.8.2 Read – HTML Form, PHP .....	297
18.8.3 Update – HTML Form, PHP .....	306
18.8.4 Login – HTML Form, PHP.....	311
18.9 Staff Module .....	313
18.9.1 Create – HTML Form, PHP.....	313
18.9.2 Read – HTML Form, PHP .....	314
18.9.3 Update – HTML Form, PHP.....	316
18.9.4 Login – HTML Form, PHP.....	318
19.0 Conclusion .....	320
19.1 Assumptions.....	320
19.2 Limitations .....	322
19.3 Enhancement.....	324
20.0 References .....	325
21.0 Appendix.....	327
21.1 Installation Guide for WampServer .....	328
21.2 Guide to Find www folder, localhost on browser, port number .....	344
21.2.1 www Folder .....	344
21.2.2 Localhost on browser.....	345
21.2.3 Port Number.....	347
21.2.4 Include Port Number in Code .....	348
21.3 Guide to PhpMyAdmin.....	349
21.3.1 Login to PhpMyAdmin .....	349
21.3.2 Create Database .....	350
21.3.3 Import Database .....	351
21.4 Minute of Meetings.....	352



### **3.0 Acknowledgement**

This project has required a significant amount of time and effort on our end. However, without our loving supervisor Miss Amira Abdul Majid's help and advice, we would not have been able to complete this software engineering project. Miss Amira Abdul Majid deserves our heartfelt gratitude. We would like to thank our parents and friends for their unwavering support and encouragement, which has greatly supported us in finishing this project.

We would also like to express our gratitude to the members of our group who worked on the project. This project was led by Teo Kai Yii, who served as the project manager. Tan Kar Wei take the role of System Analyst. The programmers of this project are Hooi Kai Jun and Sam Zheng Yew. The system tester for this project is Nisharini Ganesan.

Thank you to everyone in the group who has contributed their time and knowledge to assist us along the way.

## **4.0 Abstract**

This project is about the management system of Valley Clinic. The creation of the clinic management system is to help the clinic in supporting the daily operation that has been done manually before introducing this system. This system will not only improve the storing of data, but it will integrate well with the daily operation of the clinic. With this system, all data will be safe compared to the manual method where the data are easily being misplaced. Moreover, this system is designed not only for the patient but also the admin, doctor, and staff where each module has its known functions, uses, limitations and many more. Moreover, this documentation includes system hierarchy chart, context diagram, data flow diagram, and entity relationship diagram. These diagrams are included to show how the system would be operated. To implement this system, PHP is the selected coding language, MySQL is the selected database, Firefox is the selected browser and WampServer as the web server. To conclude, this system is created to help Valley Clinic on managing their daily operation by conducting research on other clinic management system journal article as well as making sure that the system works properly.

## 5.0 Workload Matrix

This is the workload matrix of this software development project.

ASIA PACIFIC UNIVERSITY OF TECHNOLOGY AND INNOVATION AAPP006-4-2-SDP Student Coursework Workload Matrix - Grades and Feedback Attachment							
INTAKE UCDF2005ICT(SE)		STUDENT NAME	TEO KAI YII	HOOI KAI JUN	SAM ZHENG YEW	TAN KAR WEI	NISHARINI GANESAN
Project Title: Valley Clinic Management System		TP NO.	TP058618	TP060766	TP060764	TP060354	TP059071
<b>A. Group Component</b>							
NO	ASSIGNMENT COMPONENT	ALLOCATED MARKS	CONTRIBUTION PERCENTAGE				
1	Introduction	10	20.00	20.00	20.00	20.00	100
2	Project Plan	5	50.00	50.00			100
3	System Hierarchy Chart	10	15.00	15.00	40.00	15.00	100
4	Context Diagram	5	20.00	20.00	20.00	20.00	100
5	DFD-0	5	20.00	20.00	20.00	20.00	100
6	ERD	5	15.00	15.00	40.00	15.00	100
7	Screen Design	5	20.00	20.00	20.00	20.00	100
8	Report Design	5	20.00	20.00	20.00	20.00	100
9	Test Plan	10	20.00	20.00	20.00	20.00	100
10	Significant Source Code	5	23.75	23.75	23.75	5.00	100
11	Conclusion	5	20.00	20.00	20.00	20.00	100
Total Marks and Contribution		70	22%	22%	22%	18%	16%
		Signature	Sophie	Kai Jun	Sam	Karwei	Nisha

Figure 1: Workload Matrix

## **6.0 Introduction**

### **6.1 Project Background**

As Covid-19 has affected many people's lives and business are also at the brink of closing their shops, clinics and healthcare system has become more and more important as they're the ones to treat patients and give medication, and vaccinations to patients that has been affected by the Coronavirus. Valley Clinic has also seen an uprise of patients in the couple of months as more and more of them have come to seek medications and appointments. With the vast experiences and certificates of the doctors and the Clinic has been operating for more than 20 years, more and more records and files of patients are being kept inside of a storage and therefore making it difficult for doctors to find patients details that has been a regular at the clinic and patients will need to wait for the doctor to find their records and sometimes a queue will be formed if there's a huge number of patients attending the clinic.

As so, Valley Clinic has opted to create a Management System where patients can make an appointment through the system and request a vaccination appointment as the government has allowed walk-in vaccinations at public clinics so patients can be quickly vaccinated against the coronavirus. Therefore, switching to an Online Management System will minimize the spread of Covid-19 as patients do not need to walk-in and make an appointment with a doctor and minimizing the use pen and paper to record down patient details and making it effective and efficient when doctors are searching patient records as they do not need to search it manually.

## 6.2 Problem Context

**Table: Problem Context of Valley Clinic Traditional File System (Hamel, 2018)**

Problem	Issue	Effect
No Data Backup	The paper files like patients' record can be easily destroyed in fires, floods, or other natural disasters.	The clinic will lose valuable and sensitive patients' data permanently.
Time Consuming	While patients approached to the reception for registration, the receptionist finding a patient record in a huge paper filing system might take minutes.	This may cause a long queue especially during the peak hours.
Difficulty in Modifying Data	The file system of the clinic makes it difficult for patients to update their personal details. Paper files usually don't allow data to be modified directly, the creation of new copies is required to update the existing ones.	Patients are unable to update their personal details immediately.
Data Redundancy and Data Inconsistency	When the receptionist places a patient record wrongly, or even take a patient record and did not replace it, data loss will occur and result in creating duplicate copies of patient record.	Patients' record might be loss and may result in having duplicated patients record.

No Instant Access	The clinic stores files in the clinic only. To search a patient record, the receptionist must go to the clinic in order to read the files.	Receptionist unable to access patients record virtually and instantly. This will be inconvenient when there are emergency situations.
Insufficient space and ineffectiveness	Quantity of patients' record might become overwhelming over years. Finding room to store all patients' record may cause an issue.	Filing documentations like patients' record efficiently will be difficult.
Large Queues for Walk-in Vaccination	During the wave of Covid-19, there is no proper way for patients to access to clinic. The only way for patients to get Covid Booster Vaccine is through walk-in vaccination.	The large queue outside the Valley Clinic has caused crowd. Patients was standing for hours in the queue in order to get the booster jab.
Time Restrictions in Booking Appointment	Valley Clinic only allow patients to book appointments physically or through phone call. This restricts the patients to approach or call the clinic within the business hours.	Patients can only go to the clinic or make a phone call during the business hours to make an appointment.
Low Reputation	Valley Clinic does not possess any social media. Valley Clinic only advertise through distributing business cards and brochures.	Although Valley Clinic has been operating for more than 20 years, Valley Clinic still has limited exposure to the surrounding community and youngsters. Before Covid-19 pandemic, the business of Valley Clinic had been declined constantly.

### **6.3 Proposed Solution**

According to the problem statement above, Valley Clinic is not just facing some normal issues that will be met by every organization. It seems that Valley Clinic had met some problem in managing the whole organization, however the techniques and method they use are unable to solve it perfectly. Therefore, multiples solutions are proposed by our team under the premise of solving problems of the Valley Clinic and at the same time bringing some improvement to the organization.

First, involving database management system (DBMS) is a wise choice to be done. By involving DBMS, a lot of major problem that related to files and record can be solve directly and efficiently. Data backup is very important to face some emergency situations; however, Valley Clinic does not have an actual data backup plan, using DBMS could solve this easily because in DBMS there are a backup and recovery Sfunction which enable to protect your database against data loss and reconstruct the database after any kind of data loss. DBMS is also a great way to manage and organize data, it can ensure that data would not be loss due to manmade mistake and able to increase work efficiency by using some provided functions. File systems are only able to get access by the workers of the organization, however sometimes customer also need to make changes to their personal information like phone number or house address. But database can make this happen as it can enable some part of the files be edited by authorized user. This may help a lot as the data behold of the organization are ensured to be the updated info. Overall DBMS do provide a better work environment for the organization therefore it might be a great solution to help the problems being solve.

Valley Clinic is not like the usual hospital, which is operating in a way of 24/7, therefore by using the old operating method of Valley Clinic, customer might not be able to book appointment in the situation of when the time is out of business hours. This is a serious problem which cause inconvenient to the customer and resulting that clinic losses customer bookings. Therefore, we decide to provide a platform that enable customer to book appointment whenever they want and wherever they are. These ensure that the clinic would not lose any appointment schedule and providing convenient to the customers.

Lastly, Valley Clinic had been operating for a long period therefore most of the operating method of it is still maintaining in the old traditional way. However, due to the generation change, a lot of youngsters do not notice the existence of Valley Clinic. The business of Valley Clinic had also drop due to the pandemic change. Therefore, Valley Clinic need a higher exposure to the community and get known by more people to recover their business. A good way to ensure that Valley Clinic will be known by public is to upload their information and update news to their created website or social media application like Facebook or Instagram. Providing positive and useful information on the web are a great way to promote the organization to the community and able to increase the popularity of the organization.

## **6.4 Project Aim and Objectives**

### **Project Aim**

To develop a clinic management system that will improve the workflow and management of the clinic by managing patient information and appointment.

### **Project Objectives**

1. To understand the requirements needed on controlling a clinic management system.
2. To explore on the technologies used in making the system.
3. To evaluate on how the system benefits the clinic.
4. To investigate the system, match all the set requirements.
5. To create a well-designed interface for clinic management system.
6. Able to maintain records of patience easily.
7. Easy to retrieve data from the system.
8. Able to keep tabs on the stocks of medicines.
9. Easy to generate monthly report of the clinic.

## **6.5 Project Scope**

Project scope can be defined as a set of boundaries of the project and define the goals, deadlines, and deliverables of the project. In other words, project scope can be defined as a set of requirements of the product along with the details of the work needed to achieve the outcome (Martins, 2021). For instance, in most projects, project scopes are built from scratch which usually concerns with stakeholders where these individuals will provide the requirements. These requirements will then be compiled and documented (Martins, 2021). A well-written scope statement would prove to be beneficial to the production of the project where less obstacles will be present due to the clarification on the boundaries and requirements that were previously gathered under the project scope. Along with that, project scope also binds the project team to the organization where it will provide guidance to the project team to carry out the development without a hitch. On the other hand, product scope plays an important role in the development of the proposed project. The product scope can be defined as the features of a product being manufactured. In addition, project scope plays an important role in project planning as such, an unclear scope statement would result in the project growing beyond the project team's capabilities which results in delays and setbacks in the production of the proposed project. Aside from that, a well-defined project scope would ensure that the budget and resources are sufficient to accommodate to produce the proposed project (Martins, 2021). Furthermore, it also helps reduce risk and help ensure that the proposed project meets stakeholder's expectations (Martins, 2021; Lutkevich, 2021). Some examples of project scope with regards to the proposed project would be:

- The project should be complete within RM 5,500.
- The production of the proposed project should not exceed 12 weeks.
- The testing and feedback sessions should be conducted at the end of the software development life cycle (SDLC).
- The system should be fully functional by 8 April 2022.
- The project should provide functionalities that suits the business operation of the organization.
- The project should be used by patients, staffs, doctors, and system administrator by providing functionalities for each user group.

## **6.6 Product Scope**

- The system should be accessible by any user on the World Wide Web (WWW) via web browser.
- The system should provide functionalities associated to the organization and specific target users.
- The system should allow user to register into the system or login into the system for access to the functionalities provided.
- The system should allow administrators to customize their product and services which they wish to display to customers.
- The system will be used by customers, guest, and system administrators.
- The system should provide relevant data and information requested by end users.
- The system should provide features for administrative work in terms of management.

## **7.0 Project Plan**

### **7.1 System Development Methodology**

#### **Waterfall Model**

##### **Introduction of the methodology**

Waterfall model will be the methodology used for this project. Waterfall model is a structured model that breaks down activities into a linear sequence. Each activity depends on the deliverables from the previous activity.

##### **Implementation of the methodology**

###### **Requirement Analysis**

In this stage, the requirement is studied and documented in a document. Analysis is done with the system analyst and the client so that development team can produce outcome that meets the client expectations and the goal of the project. For instance, development team will discuss with Mr. Ali about the requirement, so that the development team has better understanding of what is needed by Mr. Ali and Mr. Ali will be able to understand the limitation and flaws of several types of software.

###### **Design**

In this stage, the relationship of prerequisite with hardware and software is characterized and made to show the general framework engineering. Then the system design is then evaluated by the system analyst to ensure it meets the necessity set in the principal period of advancement process and keep away from any plan imperfections that could affect the following stages. For instance, the development team will settle on software expected to assemble the system for the physiotherapy center. From that point onward, the wireframing, UI plan, navigational design, flowchart, and Entity Relationship Diagram (ERD) are made. When these are made, the system design is investigated by the system analyst to ensure no significant blemishes that could affect the objective of the system.

## **Implementation and Unit Testing**

In this stage, the finished system design is being acknowledged to a bunch of working units. Each functioning unit is then tried to ensure it meets the necessities its determination. For instance, the developer team will code each functioning unit considering the finished working system design, then the functioning units are tried independently to ensure every one of them meets the proposed specification.

## **Integration and System Testing**

In this stage, the arrangement of working units from execution and unit testing stage is incorporated together to form a total system. The complete system is then tried to ensure it can meet the necessity set in the first phase of development. On the off chance that the system passed all the utilitarian and non-practical prerequisite tests, the documentation and system are ready to be sent to the customer. For example, the development team will carry out real certifiable information into it and run testing to ensure it can work in the genuine climate. Functional requirements, for example, login qualifications verification and non-functional requirements, for example, the response time of the site are tried to ensure it doesn't affect the customer business rule and business process. When the system finished every one of the assessments, the development team will transport the system documentation with the system to the customer and demo the system to the customer again if necessary.

## **Operation and Maintenance**

In this stage, development team will be connected by the customer to fix issues that emerged at customer side. The development team will examine the issue and fix it. The development team may likewise assist the customer with including extra highlights to stay cutthroat on the lookout or simply further develop the system security, so hackers won't get the opportunity to penetrate the system for deceitful exercises. For instance, the customer might need another page in the system to sell physiotherapy apparatuses, consequently customer mentioned this extra component to the development team and development team will sort out on planning a method for carrying out that new element in the system yet additionally not influencing the current business rule and business cycle of the customer.

## **Advantages and Disadvantages**

### **Advantages** (tutorialspoint, 2022)

- Easy to understand and explain
- Easier to control tasks
- Phases or processes are completed one after another
- Allows departmentalization and control

### **Disadvantages** (tutorialspoint, 2022)

- Not a good model for complex and object-oriented projects
- Hard to estimate time and progress
- Difficult to make changes in the middle of development.
- A lot of risks and uncertainties.

## **Justification of Chosen Methodology**

The utilization of waterfall model is to permit development team to have an exact and clear development stage to follow. On the off chance that the venture is profoundly unsurprising can be the best approach to use for system development. At the point when objectives and necessities are plainly determined, everybody in the development team will know what to do. Since each period of waterfall model is expressed clearly on what it is really going after, can be overseen effectively, and exclusion of assignments can be kept away.

Moreover, activities and outcomes are documented when using waterfall model. An excellent documentation can improve the efficiency of development team as everything that has been done and produced are documented, thus improving the efficiency of developers when they're finding the source of error to fix issues in the system. Lastly, tasks or activities are accomplished one after another, this will help the developers to focus on certain areas during the development stage and won't get distracted by other issues which may result in lower productivity and unable to finish tasks in time.

## 7.2 Project Gantt Chart

The Valley Clinic adopted Gantt Chart, which is a popular method used in project management. Gantt Chart is one of the most well-known and effective approach to show tasks against time. A list of the tasks related to each phase lie on the left side of the chart, and a proper time scale lie along the top. Each bar stands for each task, whose location and length specify the task's start dates, duration, and finish dates. The responsible member for each task will be seated beside the bars. Gantt Chart enables both project team and stakeholders to observe (Duke, 2022):

- What are the various tasks?
- When does each task start and stop?
- Where and how much do tasks depends on another tasks?
- When is the project's start and end date?

To conclude, Gantt chart shows the development process of Valley Clinic Online Management System which what needs to be done and when tasks need to be done would be displayed (Duke, 2022). The project Gantt chart for Valley Clinic Online Management System is shown below:

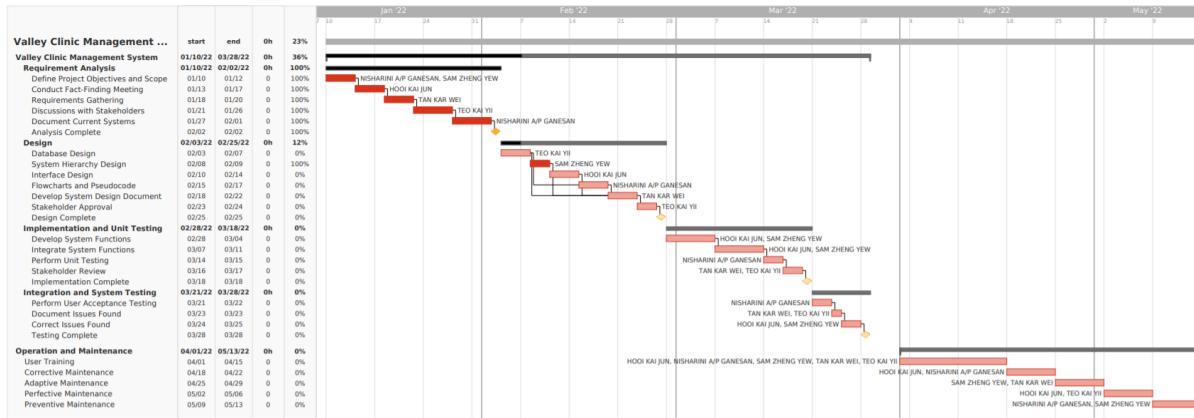


Figure 2: Project Gantt Chart

## 8.0 System Hierarchy Chart

A system hierarchy chart depicts a diagram which shows the breakdown of a system from the highest position, in this case, the proposed system, to its lowest parts ("Hierarchy Chart | MyDraw", n.d.). In other words, a hierarchy chart or diagram is a top-down modular design tool, visualized in the form of a rectangular boxes while being connected via lines ("Hierarchy Chart | MyDraw", n.d.). The lines show a connection between different modules among the system in the hierarchy chart. An organized and transparent hierarchy chart would act like a road map to direct users to various pages and information of the webpage. For instance, a system hierarchy chart is typically used in education or business environment where the aim of the hierarchy chart is to provide information which are usually hard to interpret into an easier to understand manner. The system hierarchy chart would also allow system analyst or stakeholders to observe parts of the system without having to interpret from the system directly.

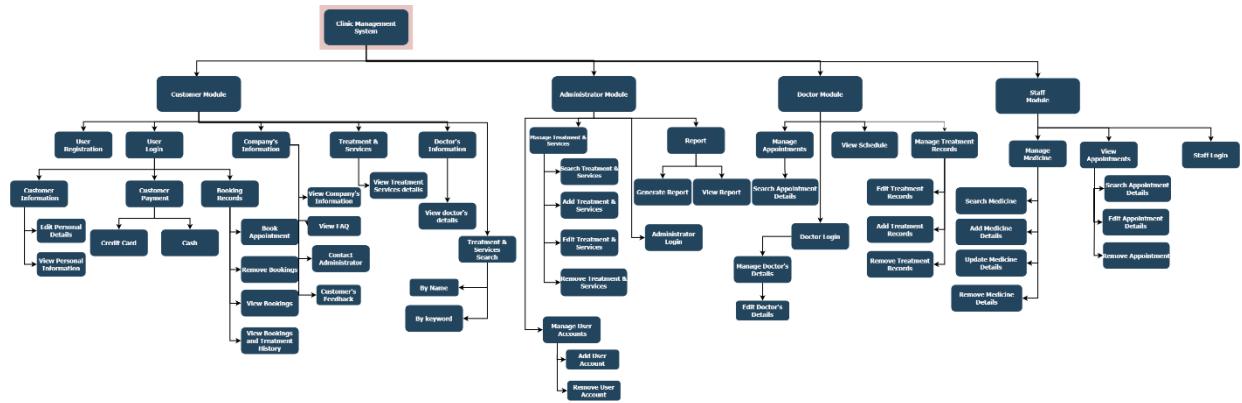


Figure 3: System Hierarchy Chart

## 9.0 Context Diagram

Context Diagram of the Valley Clinic Online Management System is designed with the concept of abstraction. The online management system is shown as a single process with the external entities of Patient, Doctor, Staff and Administrator. This context diagram depicts the whole system as a single bubble with in or out arrows which also indicates input or output data (MKS075, 2020).

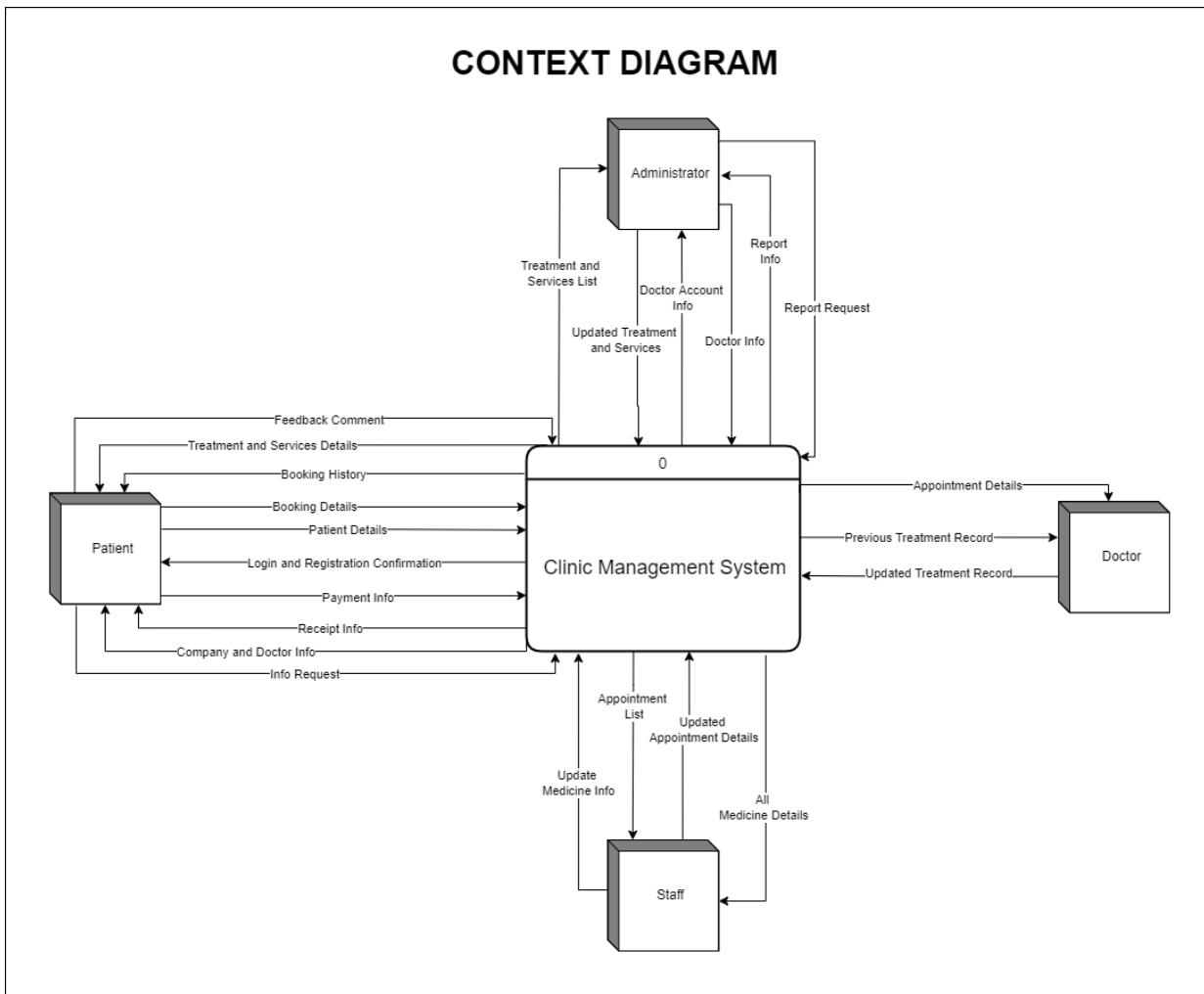


Figure 4: Context Diagram

## 10.0 Data Flow Diagrams Level 0

The Data Flow Diagram Level 0 of Valley Clinic Online Management System. The essential functions of the system are emphasized at this level. Other than that, high level process of DFD Level 0 is split down into subprocesses (MKS075, 2020).

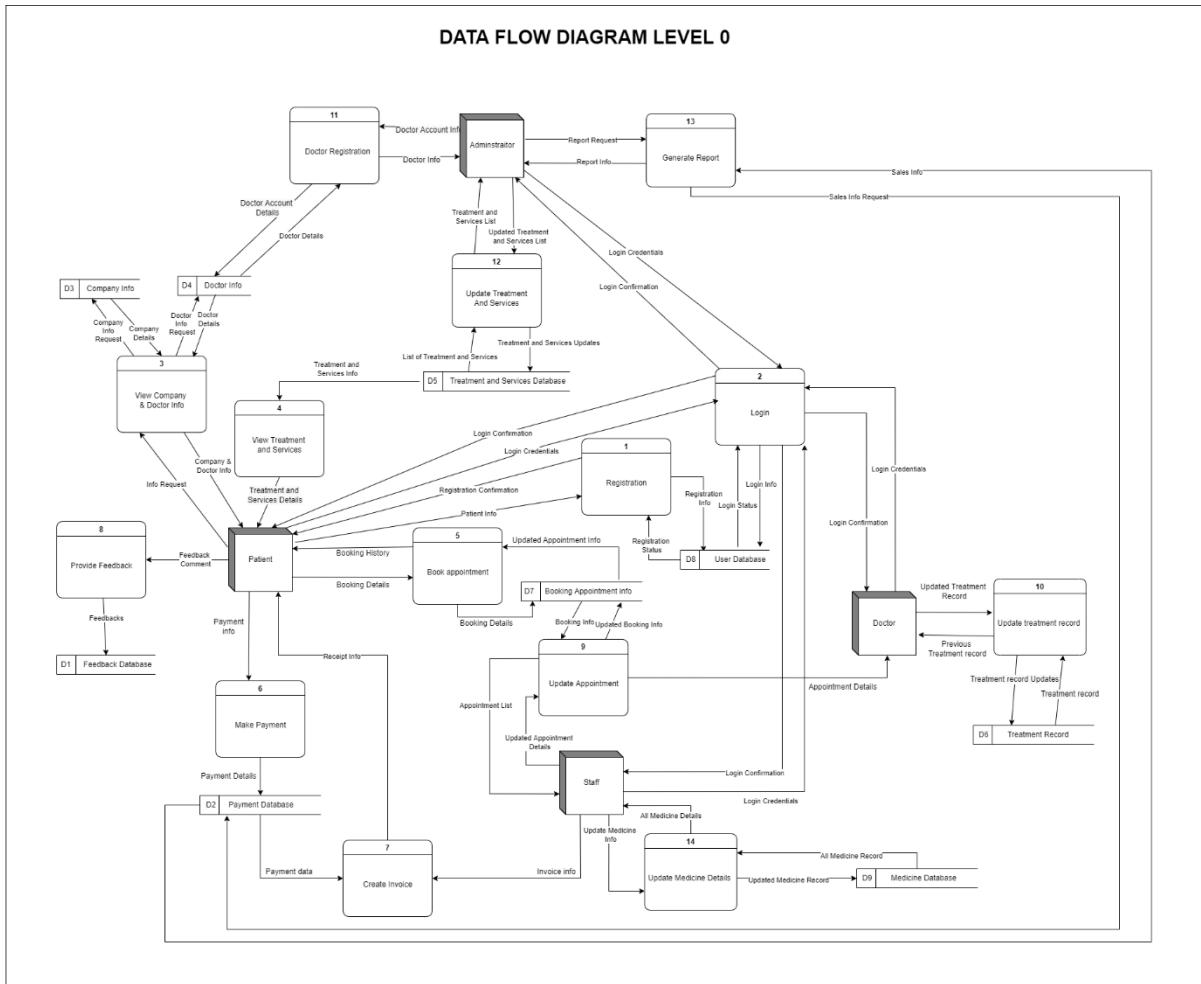


Figure 5: Data Flow Diagram Level 0

# 11.0 Data Flow Diagrams Level 1

## 11.1 Teo Kai Yee - TP058618

Valley Clinic Online Management System Data Flow Diagram Level 1 explores a little further into the DFD Level 0 elements. This diagram may be used to design or record all the required information on how the system works (MKS075, 2020).

### Process 10 Update Treatment Record

Figure below shows the data flow and sub-processes of Process 10 Update Treatment Record.

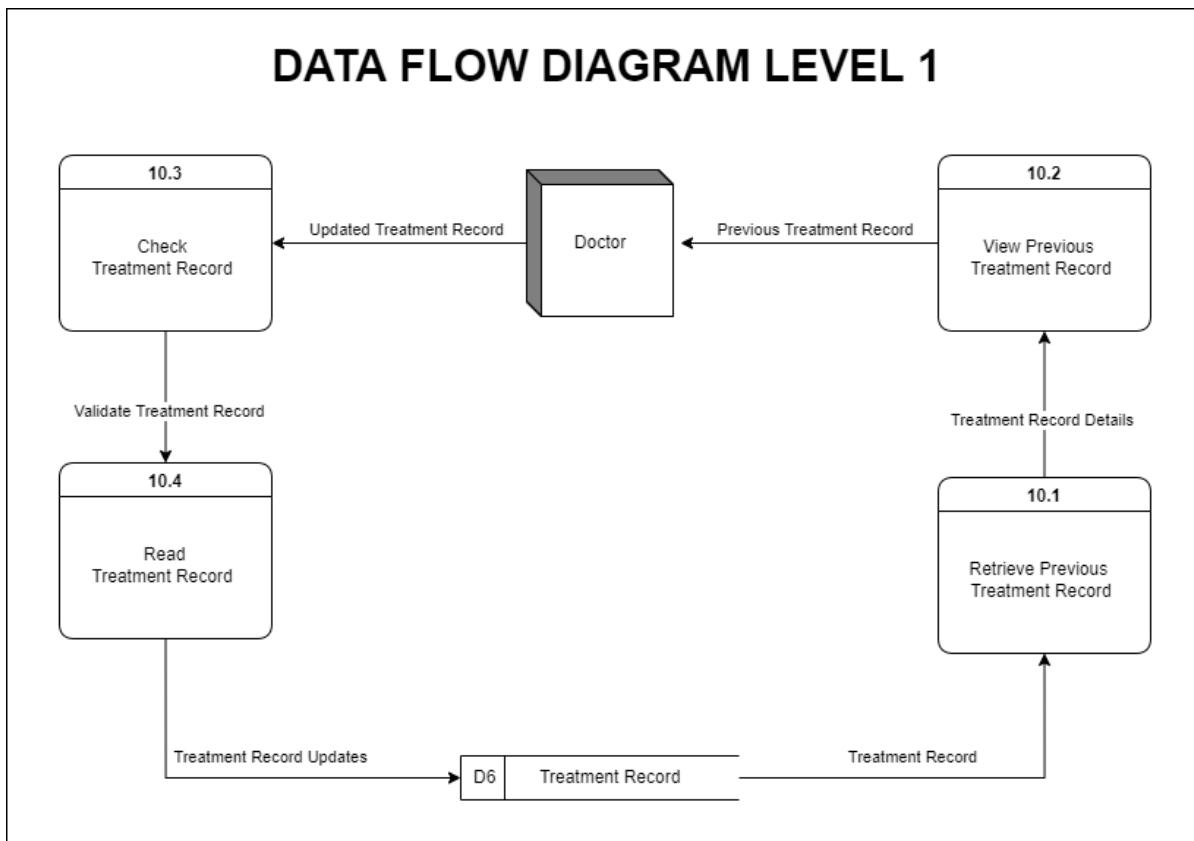


Figure 6: Data Flow Diagram Level 1 of Process 10 Update Treatment Record

## Data Flow Diagram Level 1 : Book Appointment

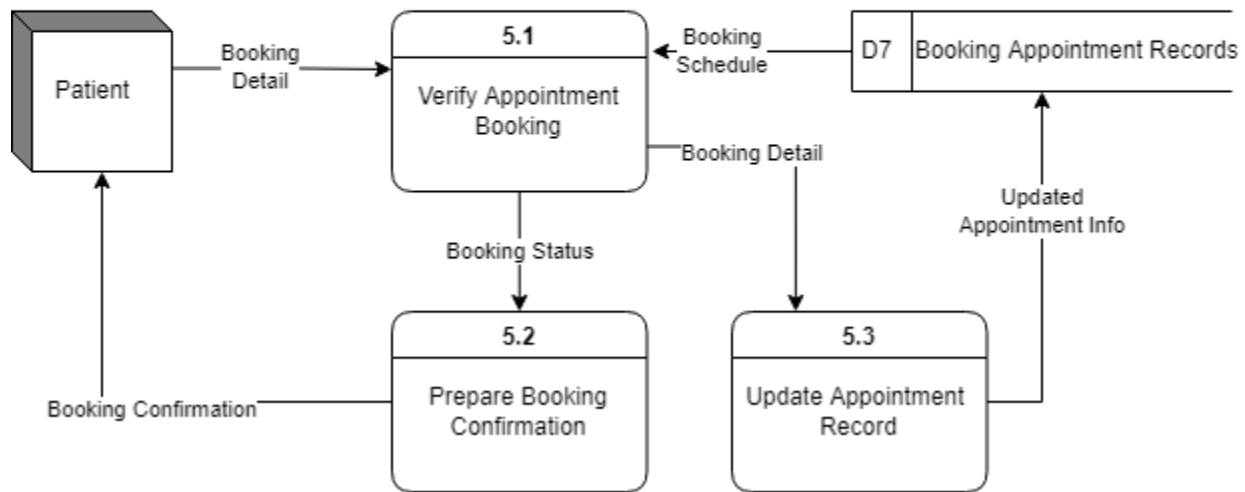


Figure 7 Data Flow Diagram Level 1: Book Appointment

### 11.3 Hooi Kai Jun – TP060766

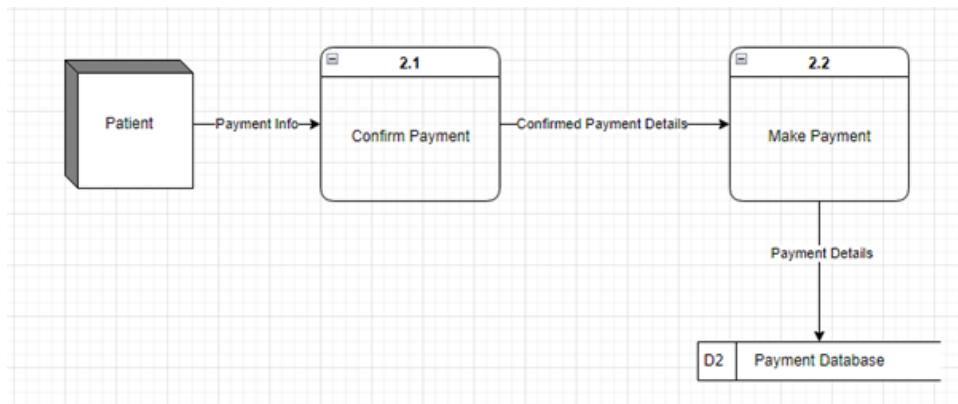


Figure 8 Data Flow Diagram Level 1: Make Payment

#### 11.4 Tan Kar Wei – TP060354

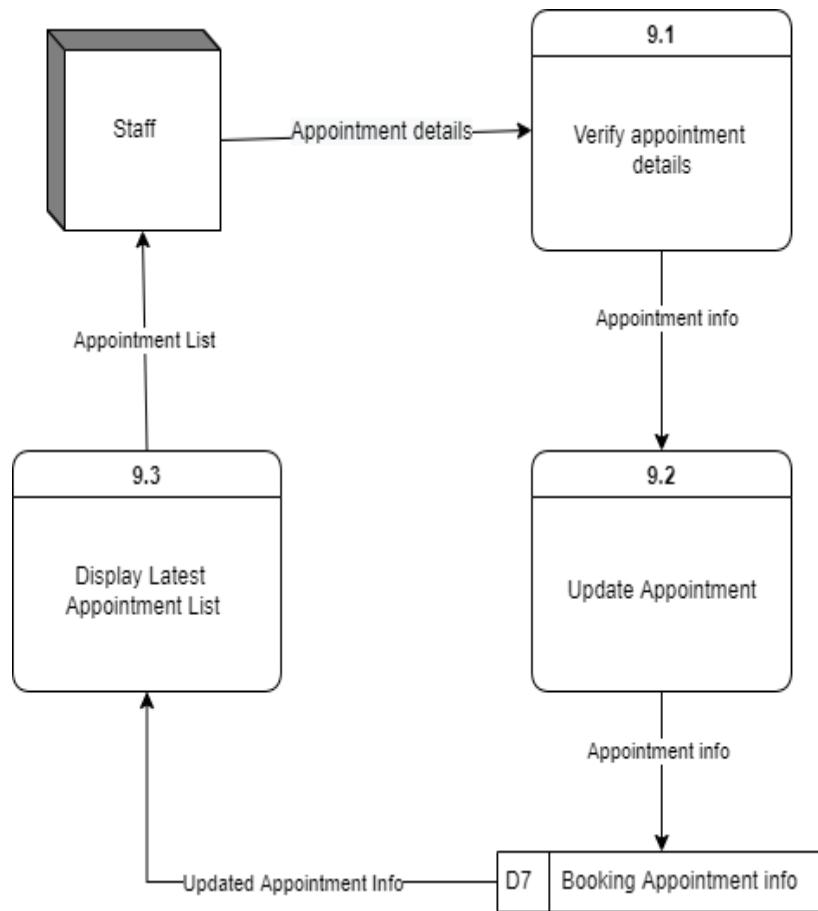


Figure 9 Data Flow Diagram Level 1: Update Appointment

## 11.5 Nisharini Ganesan – TP059071

Data Flow Diagram Level 1 for Update Medicine Information

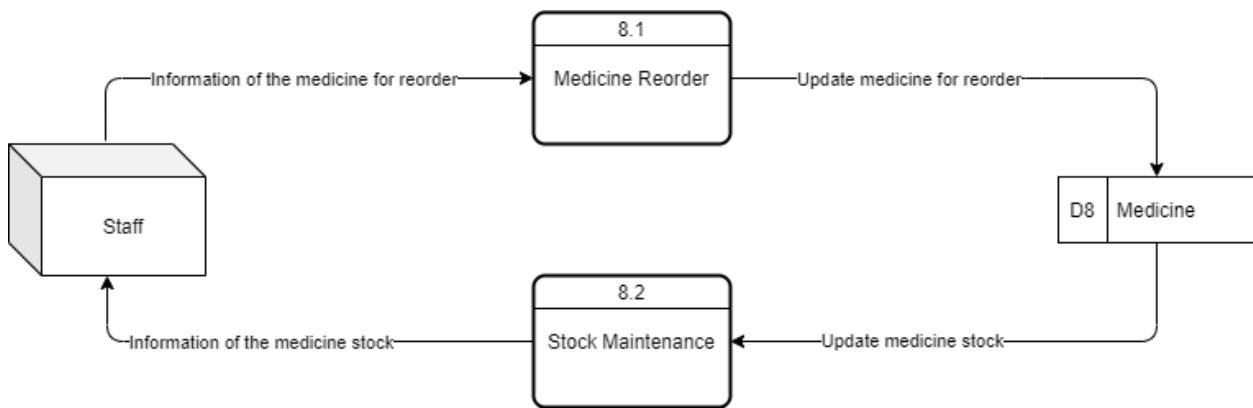


Figure 9 : Data Flow Diagram Level 1 for Update Medicine Information

## **12.0 Data Dictionary**

### **12.1 Teo Kai Yii - TP058618**

#### **Process 10 Update Treatment Record**

##### **External Entity**

Name	Doctor
Description	Doctor can login to the system by providing accurate email and password. Doctor can check appointment details in the system. Doctor will be able to view patients' treatment record and update patients' treatment record in the system.
Input Data Flows	Previous Treatment Record
Output Data Flows	Updated Treatment Record

##### **Process**

Name	10.1 Retrieve Previous Treatment Record
Description	Retrieve previous treatment record of the patient from the treatment record database.
Input Data Flows	Treatment Record
Output Data Flows	Treatment Record Details
Process Description	IF treatment record exists THEN select all treatment record ELSE Display treatment record unavailable ENDIF

Name	10.2 View Previous Treatment Record
------	-------------------------------------

Description	View previous treatment record that was stored in the system previously.
Input Data Flows	Treatment Record Details
Output Data Flows	Previous Treatment Record
Process Description	<p>IF treatment record details exist</p> <p>THEN Display all treatment record details</p> <p>ELSE</p> <p>Display treatment record details unavailable</p> <p>ENDIF</p>

Name	10.3 Check Treatment Record
Description	Check if the treatment record updated is recorded accurately.
Input Data Flows	Updated Treatment Record
Output Data Flows	Treatment Record Updates
Process Description	<p>IF fields of treatment record are filled accurately</p> <p>THEN set treatment record as validated</p> <p>ELSE</p> <p>Display Please fill up the treatment record as per requirements</p> <p>ENDIF</p>

Name	10.4 Read Treatment Record
Description	Read the validated and updated treatment record into the system.
Input Data Flows	Validate Treatment Record
Output Data Flows	Treatment Record Updates
Process Description	<p>IF treatment record is validated</p> <p>THEN Replace old treatment record with validate treatment record</p> <p>ELSE</p> <p>Display Fail to update treatment record</p> <p>ENDIF</p>

## Data Flows

.

Name	Treatment Record
Description	Data of all treatment details database.
Origin	Data Store D6 Treatment Record
Destination	Process 10.1 Retrieve Previous Treatment Record
Data Structure	ID, Doctor_ID, Patients_ID, Booking_ID, Date, Treatment_Record, Status

Name	Treatment Record Details
Description	Data of all treatment details from database.
Origin	Process 10.1 Retrieve Previous Treatment Record
Destination	Process 10.2 View Previous Treatment Record
Data Structure	ID, Doctor_ID, Patients_ID, Booking_ID, Date, Treatment_Record, Status

Name	Previous Treatment Record
Description	Data of all previous treatment details from database.
Origin	Process 10.2 View Previous Treatment Record
Destination	Doctor External Entity
Data Structure	ID, Doctor_ID, Patients_ID, Booking_ID, Date, Treatment_Record, Status

Name	Updated Treatment Record
Description	Data of updated treatment details by doctors.
Origin	Doctor External Entity
Destination	Validate Treatment Record
Data Structure	ID, Doctor_ID, Patients_ID, Booking_ID, Date, Treatment_Record, Status

Name	Validate Treatment Record
Description	Data of all validated treatment details which is accurately recorded.

Origin	Process 10.3 Check Treatment Record
Destination	Process 10.4 Read Treatment Record
Data Structure	ID, Doctor_ID, Patients_ID, Booking_ID, Date, Treatment_Record, Status

Name	Treatment Record Updates
Description	Data of the validated and updated treatment details.
Origin	Process 10.4 Read Treatment Record
Destination	Data Store D6 Treatment Record
Data Structure	ID, Doctor_ID, Patients_ID, Booking_ID, Date, Treatment_Record, Status

## Data Store

Name	D6 Treatment Record
Description	Contains all treatment data
Input Data Flows	Treatment Record Updates
Output Data Flows	Treatment Record
Data Structure	ID, Doctor_ID, Patients_ID, Booking_ID, Date, Treatment_Record, Status

## 12.2 Sam Zheng Yew - TP060764

### Process 5 Book Appointment

#### External Entity

NAME	Patient
DESCRIPTION	Patient will make booking according to the provided booking details and received a confirmation upon successfully booking an appointment.
INPUT DATA FLOWS	Booking Confirmation
OUTPUT DATA FLOWS	Booking Details

#### Process

NAME	5.1 Verify Appointment Booking
DESCRIPTION	Verify patient's booking with the booking schedule whether the slots are available and provide validation on the action
INPUT DATA FLOWS	Booking Detail, Booking Schedule
OUTPUT DATA FLOWS	Booking Status, Booking Detail
PROCESS DESCRIPTION	If booking slot are already taken/exist in booking schedule, then a validation will be generated informing that the booking was not made successfully. Otherwise, confirmation on booking status will be generated on the successfully booked appointment.

NAME	5.2 Prepare Booking Confirmation
------	----------------------------------

DESCRIPTION	Generate booking confirmation after booking had been verified on its availability.
INPUT DATA FLOWS	Booking Status
OUTPUT DATA FLOWS	Booking Confirmation
PROCESS DESCRIPTION	If booking status confirms that booking was made successful, then booking confirmation on the successfully booked appointment will be generated. Otherwise, patient will be notified that the slot was already taken/exist.

NAME	5.3 Update Appointment Record
DESCRIPTION	Update appointment details after booking had been made successfully.
INPUT DATA FLOWS	Booking Details
OUTPUT DATA FLOWS	Updated Appointment Info
PROCESS DESCRIPTION	If booking details exists, then appointment info will be updated. Otherwise, no changes will be made.

## Data Flows

NAME	Booking Detail
DESCRIPTION	Details provided by patient that is needed to book an appointment.
ORIGIN	Patient External Entity
DESTINATION	Verify Appointment Booking process
DATA STRUCTURE	Acc_ID, DocID, App_date, app_slot

NAME	Booking Status
DESCRIPTION	Confirmation on the status of patient's booking

ORIGIN	Verify Appointment Booking process
DESTINATION	Prepare Booking Confirmation process
DATA STRUCTURE	AppID, Acc_ID, DocID, App_date, app_slot, app_status

NAME	Booking Schedule
DESCRIPTION	Records of booking schedule used for verifying availability of appointment slots.
ORIGIN	Data store D7 Booking Appointment Records
DESTINATION	Verify Appointment Booking process
DATA STRUCTURE	AppID, Acc_ID, DocID, App_date, app_slot, app_status

NAME	Booking Detail
DESCRIPTION	Details of appointment made by patient to be recorded in Data store D7 Booking Appointment Records
ORIGIN	Verify Appointment Booking process
DESTINATION	Update Appointment Record process
DATA STRUCTURE	AppID, Acc_ID, DocID, App_date, app_slot, app_status

NAME	Updated Appointment Info
DESCRIPTION	Updated information on appointment made by patient.
ORIGIN	Update Appointment Record process
DESTINATION	Data store D7 Booking Appointment Records
DATA STRUCTURE	AppID, Acc_ID, DocID, App_date, app_slot, app_status

NAME	Booking Confirmation
DESCRIPTION	Processed information on booking made by patient.
ORIGIN	Prepare Booking Confirmation process
DESTINATION	Patient External Entity
DATA STRUCTURE	AppID, Acc_ID, DocID, App_date, app_slot, app_status

## Data Store

NAME	D7 Booking Appointment Records
DESCRIPTION	Contains all appointment made by patient
INPUT DATA FLOWS	Updated Appointment Info
OUTPUT DATA FLOWS	Booking Schedule
DATA STRUCTURE	AppID, Acc_ID, DocID, App_date, app_slot, app_status

## 12.3 Hooi Kai Jun - TP060766

### Process 6 Make Payment

#### External Entity

NAME	Patient
DESCRIPTION	Patient will make payment for the amount specified for their treatment that they have received.
INPUT DATA FLOWS	N/A
OUTPUT DATA FLOWS	Payment

#### Process

NAME	2.1 Confirm Payment
DESCRIPTION	Confirms the payment for the amount displayed
INPUT DATA FLOWS	Payment Info
OUTPUT DATA FLOWS	Confirmed Payment Details
PROCESS DESCRIPTION	If payment detail exists Then confirm payment Else Display payment failed End If

NAME	2.2 Make Payment
------	------------------

DESCRIPTION	Makes payment for the amount displayed
INPUT DATA FLOWS	Confirmed Payment details
OUTPUT DATA FLOWS	Payment Details
PROCESS DESCRIPTION	If payment details exists Then make payment Else Display payment failed End If

## Data Flow

NAME	Payment Details
DESCRIPTION	User payment details
ORIGIN	N/A
DESTINATION	Process 2.1 Confirm Payment
DATA STRUCTURE	AccID, CustName, Address, CardNo

NAME	Confirmed payment details
DESCRIPTION	Payment details that are confirmed
ORIGIN	Process 2.1 Confirm Payment
DESTINATION	Process 2.2 Make Payment
DATA STRUCTURE	AccID, CustName, Address, CardNo

NAME	Payment details
DESCRIPTION	User payment details are stored

ORIGIN	Process 2.2 Make Payment
DESTINATION	D2 Payment Database
DATA STRUCTURE	AccID, CustName, Address, CardNo

## Data Store

NAME	D2 Payment Database
DESCRIPTION	Contains user payment details
INPUT DATA FLOWS	Payment details
OUTPUT DATA FLOWS	N/A
DATA STRUCTURE	AccID, CustName, Address, CardNo

## 12.4 Tan Kar Wei – TP060354

### Process 9 Update Appointment

#### External Entity

Name	Staff
Description	Staff will update the appointment data based on the retrieve appointment details. After that the staff will get notify by the system about the update status whether it is successful or fail.
Input Data Flows	Appointment Update Status
Output Data Flows	Appointment Details

#### Process

Name	9.1 Verify Appointment Details
Description	Verify appointment details from booking form which fill up by patient
Input Data Flows	Appointment Details
Output Data Flows	Appointment Info
Process Description	IF appointment details verify THEN select all appointment details ELSE Display fails to verify appointment details ENDIF

Name	9.2 Update Appointment
Description	Update the appointment details retrieve earlier to the datastore
Input Data Flows	Appointment Info
Output Data Flows	Appointment info

Process Description	IF appointment info data exist THEN update data to datastore ELSE Display 'No Data Found' ENDIF
---------------------	---

Name	9.3 Display Latest Appointment List
Description	Collect all updated appointment info and display it in list form
Input Data Flows	Updated Data Status
Output Data Flows	Appointment update status
Process Description	IF Appointment info exists THEN display all appointment info in list form ELSE IF Display 'Appointment info unavailable' ENDIF

## Data Flows

Name	Appointment Details
Description	Data submitted by the patients to book their appointment
Origin	Staff
Destination	Process 9.1 Verify Appointment Details
Data Structure	App_ID, Patients_ID, Booking_ID, Date, Time, Treatment

Name	Appointment Info
Description	Data that had been verify by the staff and turning it into an actual appointment
Origin	Process 9.1 Verify Appointment Details
Destination	Process 9.2 Update Appointment
Data Structure	App_ID, Patients_ID, Booking_ID, Date, Time, Treatment, Doctor_ID

Name	Appointment Info
Description	Data about actual appointment that need to be save into the database
Origin	Process 9.2 Update Appointment
Destination	Data Store D7 Booking Appointment Info
Data Structure	App_ID, Patients_ID, Booking_ID, Date, Time, Treatment, Doctor_ID

Name	Updated Appointment Info
Description	Updated appointment info which stores in database
Origin	Data Store D7 Booking Appointment Info
Destination	Process 9.3 Display Latest Appointment List
Data Structure	App_ID, Patients_ID, Booking_ID, Date, Time, Treatment, Doctor_ID

Name	Appointment List
Description	Data of all appointment info which is in list form.
Origin	Process 9.3 Display Latest Appointment List
Destination	Staff External Entity
Data Structure	App_ID, Patients_ID, Booking_ID, Date, Time, Treatment, Doctor_ID

## Data Store

Name	D7 Booking Appointment Info
Description	Contains All Appointment Data
Input Data Flows	Appointment Info
Output Data Flows	Updated Appointment Info
Data Structure	App_ID, Patients_ID, Booking_ID, Date, Time, Treatment, Doctor_ID

## 12.5 Nisharini Ganesan – TP059071

### Process 8 Update Medicine Information

#### External Entity

<b>Name</b>	Staff
<b>Description</b>	The staff of Valley Clinic will manage and update the stock of the medicines in the clinic.
<b>Input Data Flows</b>	Information of the medicine stock.
<b>Output Data Flows</b>	Information of the medicine for reorder.

#### Process

<b>Name</b>	<b>8.1 Medicine Reorder</b>
<b>Description</b>	Overlook the reorder of medicine.
<b>Input Data Flows</b>	Information of the medicine for reorder.
<b>Output Data Flows</b>	Update medicine for reorder.
<b>Process Description</b>	If Medicine does not exist  Then create order  Else  Display order failed  Endif

<b>Name</b>	<b>8.2 Stock Maintenance</b>
<b>Description</b>	Overlook the stocks of medicine before and after ordering.
<b>Input Data Flows</b>	Update medicine stock.
<b>Output Data Flows</b>	Information of the medicine stock.
<b>Process Description</b>	<p>If Stock exist</p> <p>    Then check stock balance</p> <p>    Else</p> <p>        Display checking failed</p> <p>    Endif</p>

## Data Flows

<b>Name</b>	<b>Information of the medicine for reorder</b>
<b>Description</b>	Basic information needed to make the order of the medicines needed.
<b>Origin</b>	Staff
<b>Destination</b>	Process 8.1 Medicine Reorder
<b>Data Structure</b>	MedID, OrderID.

<b>Name</b>	<b>Update medicine for reorder</b>
<b>Description</b>	Update the system on the orders made for the needed medicines.
<b>Origin</b>	Process 8.1 Medicine Reorder
<b>Destination</b>	D8 Medicine
<b>Data Structure</b>	MedID, OrderID.

<b>Name</b>	<b>Update medicine stock</b>
<b>Description</b>	Update on the number of stocks after receiving the latest order of medicines.
<b>Origin</b>	D8 Medicine
<b>Destination</b>	Process 8.2 Stock Maintenance
<b>Data Structure</b>	MedID, StockID.

<b>Name</b>	<b>Information of the medicine stock</b>
<b>Description</b>	Keep all information of the medicines that are in stock.
<b>Origin</b>	Process 8.2 Stock Maintenance
<b>Destination</b>	Staff
<b>Data Structure</b>	MedID, StockID

## Data Store

<b>Name</b>	<b>D8 Medicine</b>
<b>Description</b>	Contains details on all medicines.
<b>Input Data Flows</b>	Update medicine for reorder
<b>Output Data Flows</b>	Update medicine stock
<b>Data Structure</b>	MedID, OrderID, StockID.

## 13.0 Entity Relationship Diagram

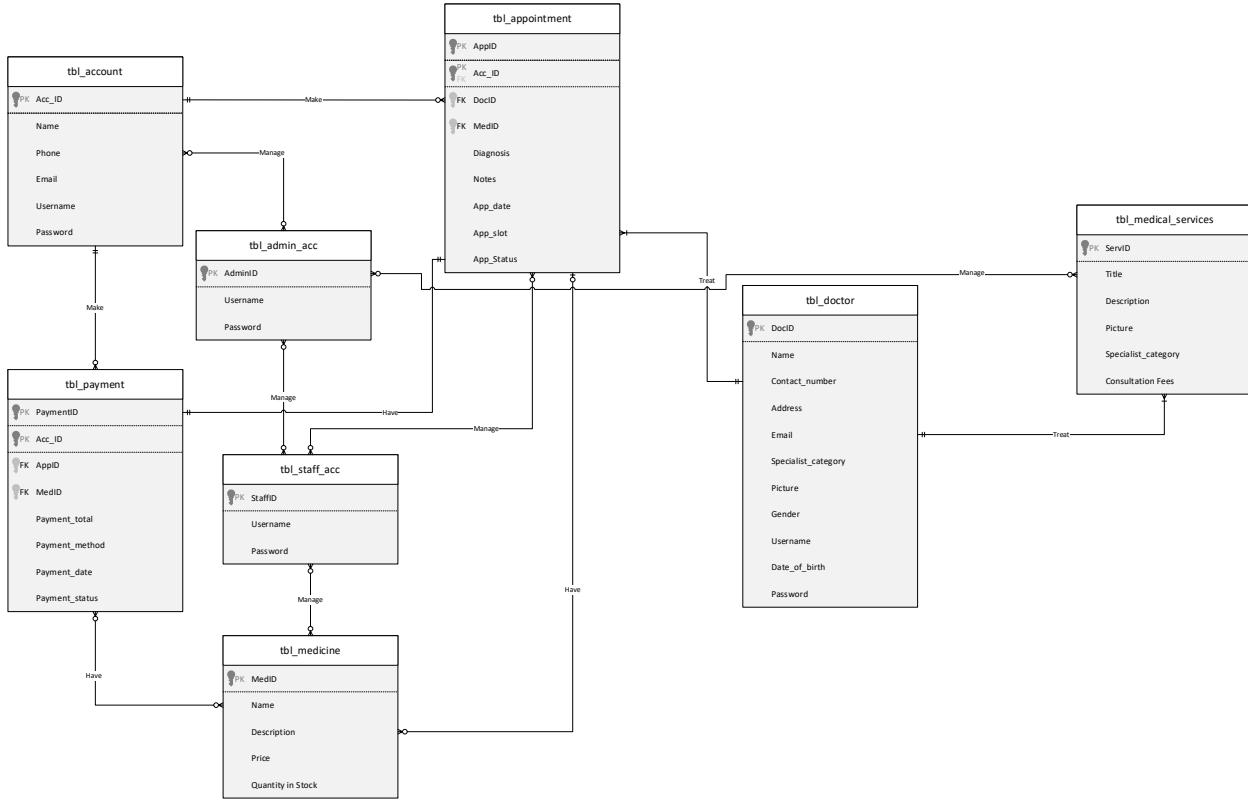


Figure 10 Entity Relationship Diagram (Valley Clinic)

Based on the illustration above in Figure 7, the diagram depicts an entity relationship diagram design for the backend use of Valley Clinic database. First and foremost, an entity relationship diagram can be defined as a graphical representation that depicts relationships among people, objects, places or rather entities that are stored in a database. Any instances of entities in this context are considered as objects and are data which are stored as a singular entity. An entity relationship diagram is most commonly used in the planning stage of a software project which helps to identify different system elements and their relationships with each other. Based on the illustrated ER diagram above, shows the relationship of each individual entity with each other, for example, in this case, how each entity is connected and how the system information is produced through the defined relationships. In the ERD, multiple symbols are being used which consist of an entity, attributes, cardinality, and connecting lines. An entity is an object or concept where information is stored, for example, the user of Valley Clinic will have to register their personal

account via credentials and personal information which will be stored in an entity called `tbl_account`. Besides the entity, there are also other elements such as attributes which is a property, traits, or characteristics of an entity they are related to which are distinguishing characteristics of the entity. For example, in the entity `tbl_account`, consist of attributes such as `AccID`, `Phone`, `Email`, `username` and `password` these are the attributes or key information that are related to the entity they are in which in this case and the information that resides in that particular entity, `tbl_account`. Aside from that, an entity relationship diagram can consist of multiple entities which in this case has to be connected with each other, the connecting lines along with the cardinality specifies the relationship that they are in and the action they perform along with specifying how many instances of an entity relate to one instances of another entity. This is called cardinality, where it specifies the maximum number of relationships or minimum number of relationships between the two entities whereas the action shows how two entities share information in the database. (Entity Relationship Diagram, n.d.); (Ultimate Entity Relationship Diagram Tutorial (ER Diagrams), 2022).

ER diagram provides a lot of benefits as a framework for creating and manipulating database (Ultimate Entity Relationship Diagram Tutorial (ER Diagrams), 2022). Some of the benefits of an entity relationship diagram is that ER diagram are easy to understand. For example, any individuals will be able to work with ER diagram easily without having to go through extensive training as ER diagram are easy to perceive and use in delivering information to other parties. Aside from that, ER diagrams are also readily translatable into relational tables due to the design of an ERD being already in a favourable form which allows the user to easily transfer the information in an ER diagram into a database. (Ultimate Entity Relationship Diagram Tutorial (ER Diagrams), 2022)

## 14.0 Pseudocode or Flowcharts

### 14.1 Teo Kai Yii – TP058618

#### Process 10 Update Treatment Record

The flowchart below describes the flow of doctor updating the treatment record.

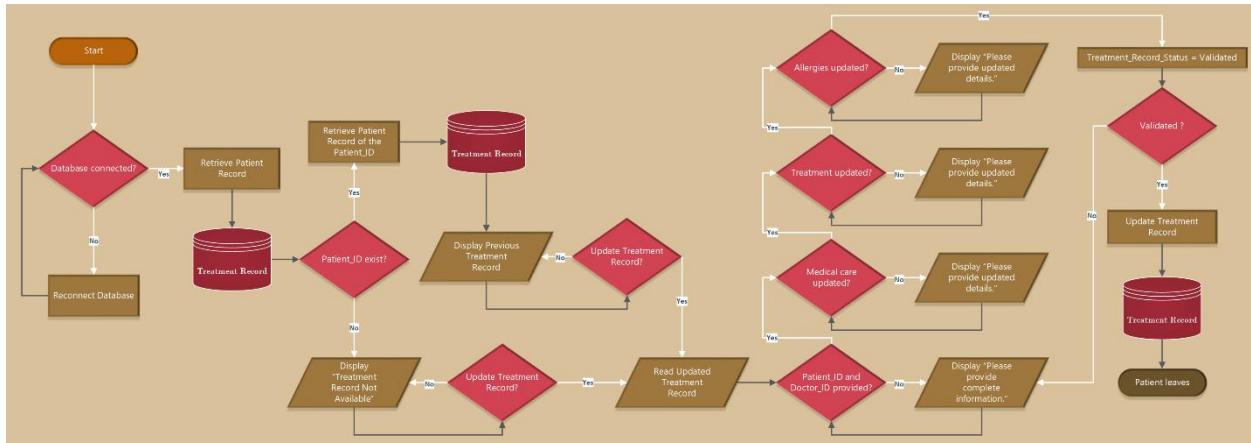


Figure 11: Flowchart for Process 10 Update Treatment Record

## Pseudocode & Flowchart: Book Appointment

```
BEGIN
    User sign-in
    Display homepage
    Display Book appointment page
    Select doctor
    read DocID
    Display Available Slots
    Enter appointment date
    Read App_date
    Enter appointment time
    Read App_slot
    Select "Book" button
    IF (App_date already exist AND App_slot already exist AND
        already exist DocID) THEN
        UPDATE to database
        Display "Booking had been made successfully!"
    ELSE
        Display "Booking already exist! Please choose a
        different slot"
        Display Book appointment page
    ENDIF
END
```

Figure 12: Pseudocode: Book Appointment

## Flowchart

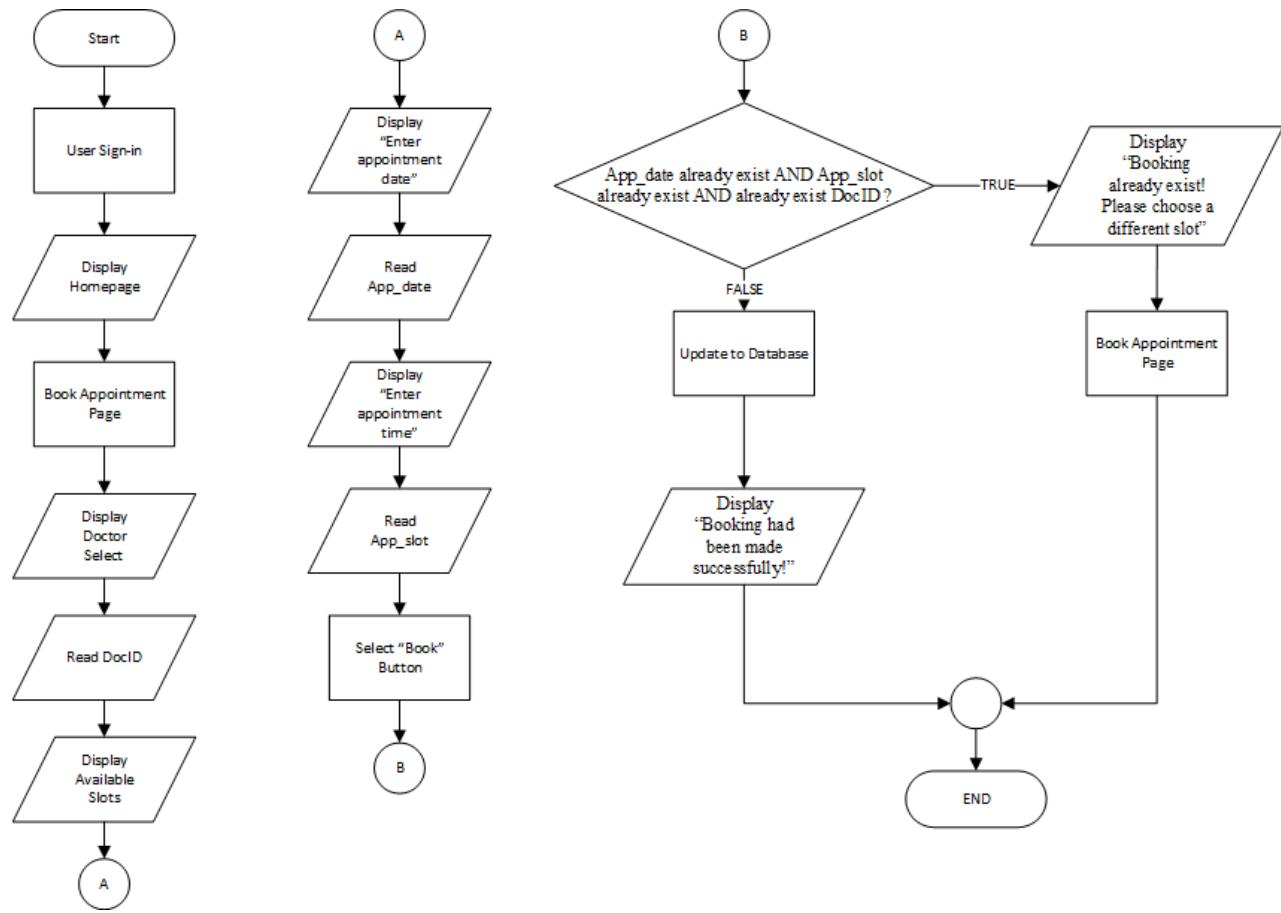


Figure 13: Flowchart: Book Appointment

## 14.3 Hooi Kai Jun – TP060766

### Payment flowchart

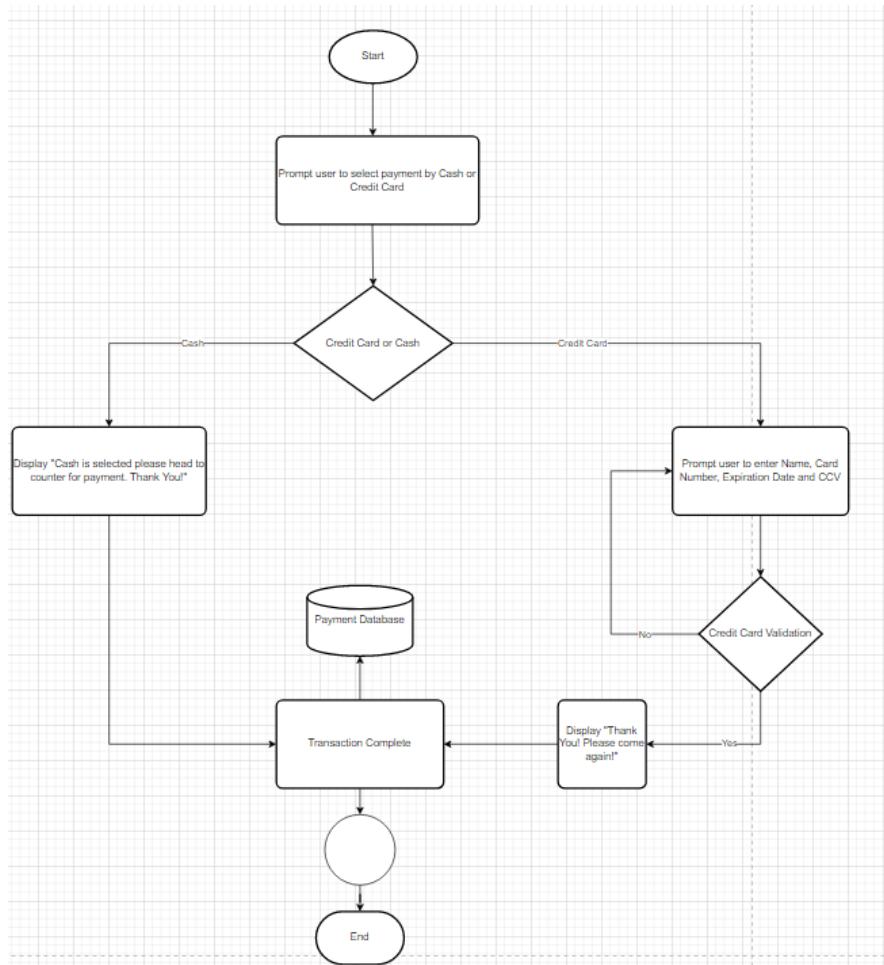


Figure 14: Payment Flowchart

#### **14.4 Tan Kar Wei – TP060354**

Pseudocode (Update Appointment)

```
BEGIN
    Staff sign in
    Display Staff Homepage
    Display Update Appointment Page
    Select Appointment
    Read App_ID
    Display Available Doctor
    Enter Doctor_ID
    Read Doctor_ID
    Select 'Update' Button
    Update data to database
    Display 'Data Update Successfully'
    Read data from database
    Display Latest Appointment List
END
```

*Figure 15: Pseudocode of Update Appointment*

## 14.5 Nisharini Ganesan - TP059071

Flow Chart for Update Medicine Information

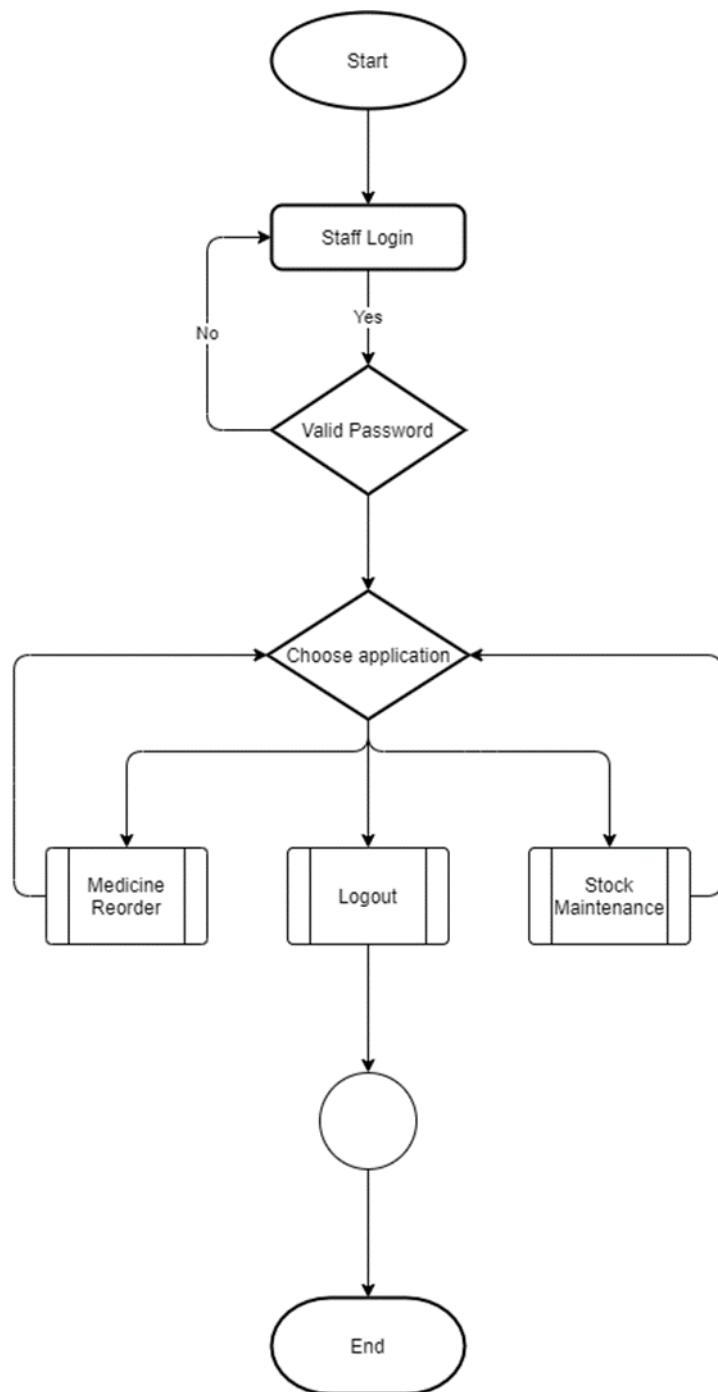
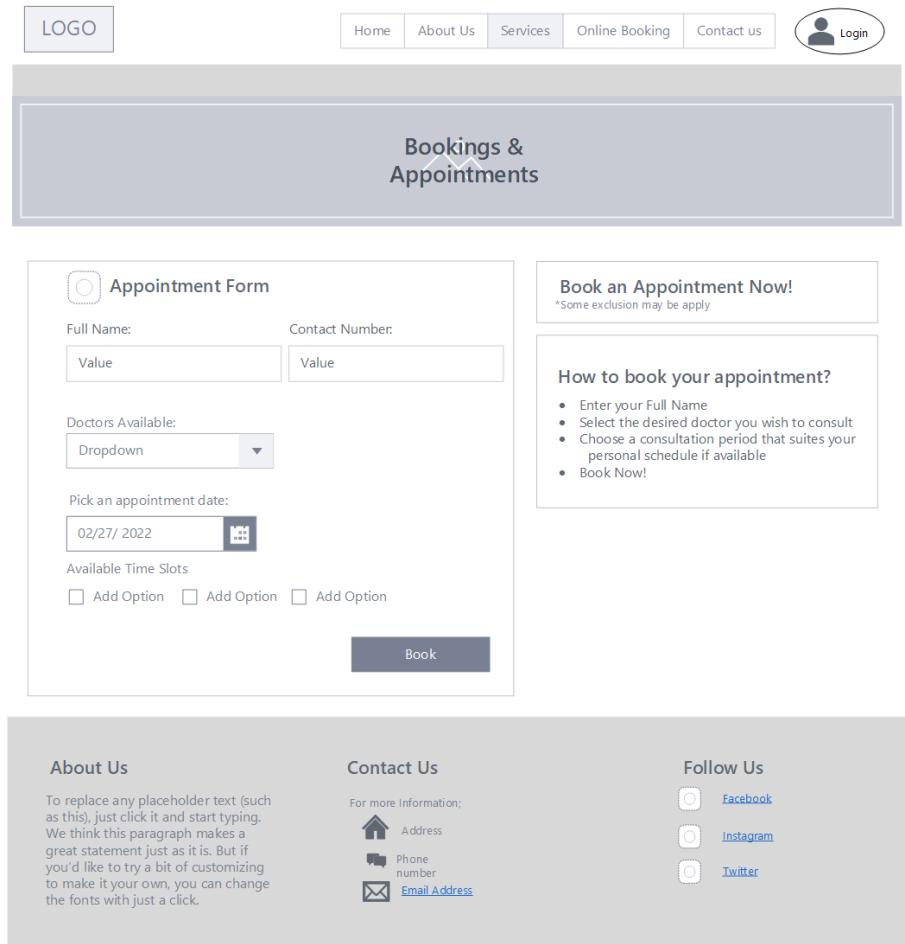


Figure 16 : Flowchart for Update Medicine Information

# 15.0 Screen Design and User Manual

## 15.1 Customer Module

### 15.1.1 Customer Booking



The wireframe illustrates the layout of the Customer Appointment Booking page. At the top, there is a header bar with a logo on the left, followed by navigation links: Home, About Us, Services, Online Booking, Contact us, and a Login button. Below the header, a main title 'Bookings & Appointments' is centered. The main content area is divided into two columns. The left column contains an 'Appointment Form' section with fields for Full Name (with a placeholder 'Value') and Contact Number (with a placeholder 'Value'). It also includes a dropdown menu labeled 'Doctors Available' with 'Dropdown' as the placeholder, a date input field showing '02/27/2022', and a section for 'Available Time Slots' with three checkboxes labeled 'Add Option'. A large 'Book' button is located at the bottom of this column. The right column features a section titled 'Book an Appointment Now!' with a note '(Some exclusion may be apply)'. Below this is a 'How to book your appointment?' section containing a bulleted list of steps: Enter your Full Name, Select the desired doctor you wish to consult, Choose a consultation period that suits your personal schedule if available, and Book Now! At the bottom of the page, there are three footer sections: 'About Us' (placeholder text about replacing placeholder text), 'Contact Us' (placeholder text for address, phone number, and email address), and 'Follow Us' (social media icons for Facebook, Instagram, and Twitter).

Figure 10 Customer Appointment Booking

As shown in Figure 10, the wireframe design of Valley Clinic Management System compose my three main components namely, the header, the body content, as well as a footer. Similarly shown in other webpages, the customer booking page provides navigation through the use of navigation bar which the user can opt for navigating to other webpages associated to the user group. If the user would like to cancel their booking and return to the homepage, they can do so by clicking on the nav bar “home” tab. In the main body of the webpage, customers will be able to make their booking based on their preference by entering their personal information such as contact number, and full name as well as other required information needed for the booking, for

instance, the user will have to choose the doctor which they wish to visit on the date of the appointment as well as the date and time slots they wish to book.

### 15.1.2 Customer Confirm Booking Details

The screenshot shows a web-based booking application interface. At the top, there's a navigation bar with a logo on the left and links for Home, About Us, Services, Online Booking, and Contact us. To the right of these is a circular 'Login' button with a user icon. Below the navigation is a main title 'Bookings & Appointments' and a sub-section title 'Confirm Appointment Details'. The form itself has several input fields: 'Full Name' and 'Contact Number' each with a 'Value' placeholder; a 'Doctors Available' dropdown menu showing 'Dropdown'; a date input field set to '02/27/ 2022' with a calendar icon; and a 'Available Time Slots' section containing three checkboxes labeled 'Add Option'. At the bottom right of the form is a dark blue 'Next' button.

Figure 11 Customer Confirm Booking Details

Based on Figure 11 shown the confirm booking page for customer viewing pleasure. In this webpage, the user will be given details from the previously entered form and the information will be used by the user for validation. If the information on displayed are not as user's liking, they are free to make any changes until their heart is content. After which, the user can submit the form for the last time before heading into the payment page.

### 15.1.3 Payment Gateway

The figure consists of three main sections. The top section shows a 'Payment' screen with a 'Cash' button, a 'Total Amount Due:' field containing 'Value', and a note about invoices being sent via email. The middle section shows a 'Credit Card' payment screen with fields for 'Card Number', 'Name on Card', 'Expiry Date', and 'Security Code'. The bottom section is a footer area with three columns: 'About Us' (placeholder text), 'Contact Us' (with icons for address, phone number, and email address), and 'Follow Us' (with icons for Facebook, Instagram, and Twitter).

**About Us**  
To replace any placeholder text (such as this), just click it and start typing. We think this paragraph makes a great statement just as it is. But if you'd like to try a bit of customizing to make it your own, you can change the fonts with just a click.

**Contact Us**  
For more information:  
 Address  
 Phone number  
 Email Address

**Follow Us**  
 [Facebook](#)  
 [Instagram](#)  
 [Twitter](#)

Figure 12 Customer Payment

In the Customer Payment Page of which previously originate from confirm booking page, the user can opt to make payment via various payment method such as credit card or by cash. If the user chooses the latter option, they will be directed to the payment gateway of which information regarding the payment is shown. Similarly, if the customer chooses to pay with credit card, they will be redirected to a new form which requires the user to enter their credit card information.

### 15.1.4 Customer Homepage



Figure 17 Valley Clinic Homepage

Figure 15 shows the screen design of Valley Clinic webpage which is accessible by various user group supported by the Clinic Management System. Upon visiting the webpage, the user will be prompt into the main page of the Clinic. In the homepage, user is bombarded with multiple elements, each performing their specific actions, for example, in the header of the webpage, user is given validation that they are indeed in the official webpage of Valley Clinic, given by the logo

located on the top-left of the display. This will provide a sense of relief to the user to know that they are not accessing any dangerous webpage. Aside from that, the webpage's header also includes a horizontal navigational bar which act as a directory for the user if they wish to navigate across particular section of the system. For example, user can navigate to specific pages provided by the proposed system given it being the homepage, appointment booking, company background, details of available services and treatments, and even the contact us page where user can notify the clinic staff of any enquiry. Back to the homepage, located on the first section of the page consist of a carousel which provide a slideshow of information depicting related information in terms of healthcare which include an interactive element which lets user navigate through the display according to their preference. Below of which locates a brief detail explanation on the overview of the overall business operation of Valley Clinic and their services. Furthermore, an array of information regarding services highlights and medical condition process are also provided in the homepage of the proposed system. Lastly, a familiar footer which consist of a brief description of the company, contact information of the clinic itself, and social links which are accessible to the user signifies the end of the homepage.

## 15.1.5 About Us

LOGO

Home   About Us   Services   Online Booking   Contact us

Login

**About Us**  
To replace any placeholder text (such as this), just click it and start typing. We think this paragraph makes a great statement just as it is. But if you'd like to try a bit of customizing to make it your own, you can change the fonts with just a click.

**Our Mission**  
To replace any placeholder text (such as this), just click it and start typing. We think this paragraph makes a great statement just as it is. But if you'd like to try a bit of customizing to make it your own, you can change the fonts with just a click.

**Our Vision**  
To replace any placeholder text (such as this), just click it and start typing. We think this paragraph makes a great statement just as it is. But if you'd like to try a bit of customizing to make it your own, you can change the fonts with just a click.

**About Us**  
To replace any placeholder text (such as this), just click it and start typing. We think this paragraph makes a great statement just as it is. But if you'd like to try a bit of customizing to make it your own, you can change the fonts with just a click.

**Contact Us**  
For more Information;  
 Address  
 Phone number  
 Email Address

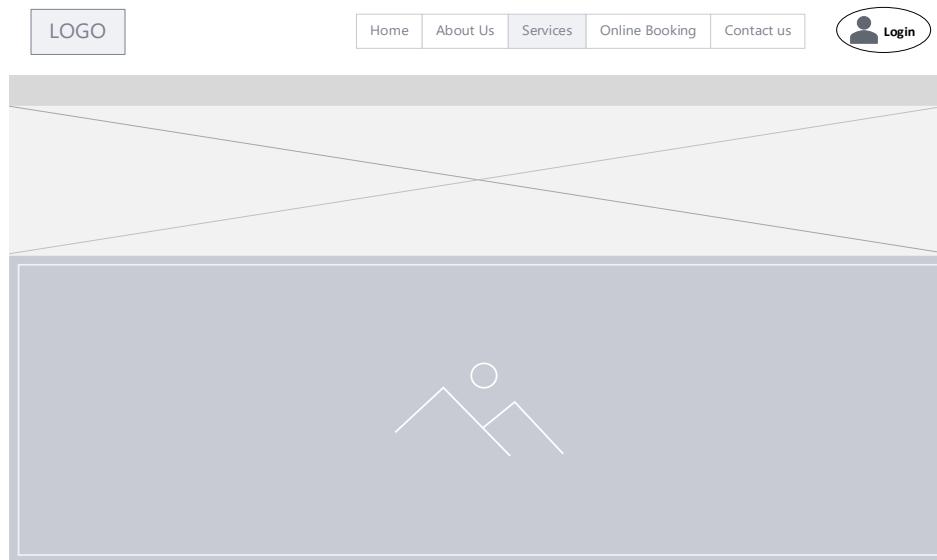
**Follow Us**  
 [Facebook](#)  
 [Instagram](#)  
 [Twitter](#)

Figure 13 Valley Clinic About Us page

Moving into the about us page of Valley Clinic, which is portrait in Figure 16, a sense of familiarity can be seen throughout the webpages where a consistent layout is used, for example, the header of the webpage consist of the company's logo and a navigational bar and also a login or register button where user can interact with to perform specific functions. Aside from that, the content of the about us page as the name implies consist of a brief introduction the company which

can be viewed by visitors and also information on their mission and vision. Lastly, the final element which is included into the about us page would be the footer, which can be seen similar to the homepage. The footer provides similar function and layout of information as seen in the homepage of the proposed system.

## 15.1.6 Contact Us



The contact us page features a logo placeholder, a navigation bar with Home, About Us, Services, Online Booking, and Contact us links, and a login button. The main content area includes a stylized mountain icon and a 'Get in Touch With Us!' section with social media links, contact form fields, and a comment area.

**Get in Touch With Us!**

Our Socials	Contacts	Location
<a href="#">Facebook</a>	Phone Number	Address
<a href="#">Instagram</a>	<a href="#">Email</a>	
<a href="#">Twitter</a>		

**Contact Us**

Name	Contact Number
Enter Text	Enter Text

Email

Enter Text
------------

Comment

Enter Text
------------

**About Us**

To replace any placeholder text (such as this), just click it and start typing. We think this paragraph makes a great statement just as it is. But if you'd like to try a bit of customizing to make it your own, you can change the fonts with just a click.

**Contact Us**

For more Information:

Address
Phone number
Email Address

**Follow Us**

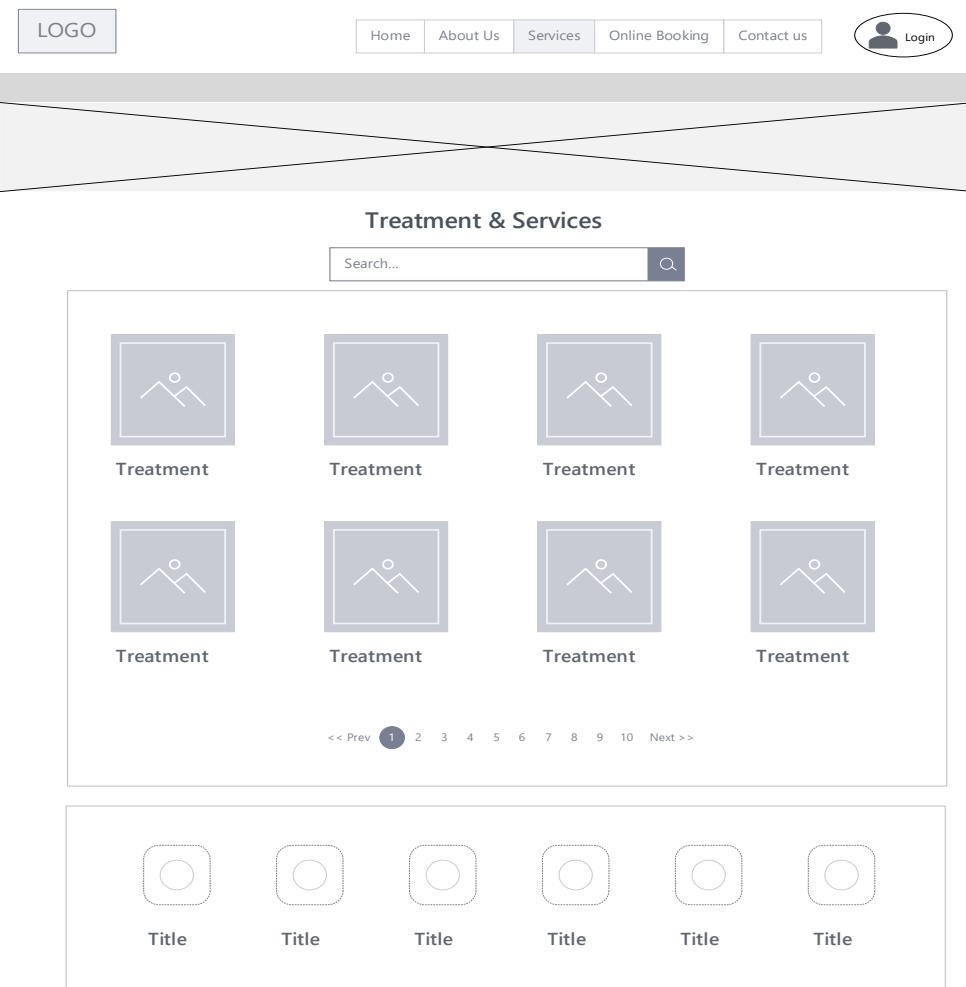
<a href="#">Facebook</a>
<a href="#">Instagram</a>
<a href="#">Twitter</a>

Figure 19 Valley Clinic Contact Us page

As shown in Figure 17, the webpage of Valley Clinic also includes a contact us page where user can opt to contact the company for any enquiry that they have. Similar to present webpages

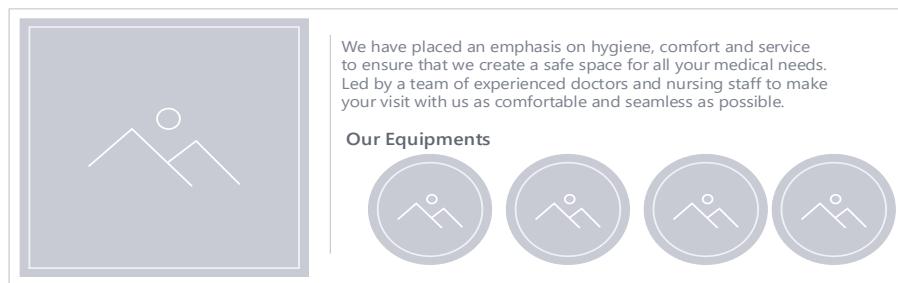
in the proposed system, the contact us page is no difference where the header of the page consists of a navigational bar as well as a login option. Below the option, a separator is used to signify the next content of the webpage where a representation of the physical location is up on display. A google map extension is used to display the location of which the clinic is located physically where the user can interact with to further investigate the surrounding areas. Furthermore, three hyperlinks which redirects user to the company's Facebook page, Instagram, as well as Twitter page are also displayed on the same page. Next to it would be the clinic's phone number as well as email address where user can opt to contact their staff for any enquiry that they wish to be answered. Lastly, there is also the physical address where user can use to visit the physical shop if opt to. A section of the page is reserved for a contact us feature, where the user can enter any messages, they wish to be replied to either it being a suggestion or an array of enquiries that are needed attention of staff members. Multiple columns are to be filled such as the name of the sender, email address, phone number as well as any comment they wish to include before sending the ticket for staffs to review. Lastly, the footer section is added into the contact us page which is used to signify the end of the page as well as displaying important information to the user.

## 15.1.7 Treatment And Services



The wireframe shows a header with a logo, navigation links (Home, About Us, Services, Online Booking, Contact us), and a login button. Below the header is a large gray banner area. The main content area is titled "Treatment & Services" and contains a search bar. A grid of 8 treatment icons (each showing a stylized mountain range with a circle) is displayed in two rows of four. Below the grid is a pagination control with links for << Prev, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, Next >>. At the bottom, there's a row of six placeholder cards labeled "Title".

### Services & Equipments



This wireframe shows a large image of a stylized mountain range on the left. To its right is a text block about hygiene and service, followed by a section titled "Our Equipments" with four circular icons. The entire section is contained within a light gray box.

### About Us

To replace any placeholder text (such as this), just click it and start typing. We think this paragraph makes a great statement just as it is. But if you'd like to try a bit of customizing to make it your own, you can change the fonts with just a click.

### Contact Us

For more Information;  
 Address  
 Phone number  
 Email Address

### Follow Us

 Facebook  
 Instagram  
 Twitter

*Figure 20 Valley Clinic Treatment and Services page*

As shown in Figure 18, the wireframe depicts the layout of Valley Clinic Treatment and Services webpage. The above webpage provides an overview of the available treatments as well as services provided by the business organization. For example, the layout of the webpage consists of a header which can be seen consistent in multiple webpages, the footer, and the main content. First and foremost, the header of the webpage consists of a navigation bar and login function which can be seen across multiple webpages. Aside from that the main focus of the webpage which is the available treatment and services are displayed in the form of text and images located in the main body of the HTML page. User can also opt to search the available treatment provided by the clinic with the help of the search feature. Aside from that, a brief description on the provided services and the equipment used are also shown to provide insight to the user on the services that are to be expected. Lastly, the footer which can be seen consistent on all webpages are also included in the treatment and services webpage located at the end of the webpage.

## 15.1.8 Medical Condition

LOGO

Home About Us Services Online Booking Contact us Login

## Medical Condition

Description

To replace any placeholder text (such as this), just click it and start typing. We think this paragraph makes a great statement just as it is. But if you'd like to try a bit of customizing to make it your own, you can change the fonts with just a click.

Consultation Fees

RM ... - RM ... Per Consultation

Specialist Doctor

About Us

To replace any placeholder text (such as this), just click it and start typing. We think this paragraph makes a great statement just as it is. But if you'd like to try a bit of customizing to make it your own, you can change the fonts with just a click.

Contact Us

For more Information;

- Address
- Phone number
- Email Address

Follow Us

- Facebook
- Instagram
- Twitter

Figure 21 Valley Clinic Treatment Details page

In order to proceed to the webpage shown in Figure 19, user would have to access the previous webpage associated to the treatment and services webpage which is accessible through the navigation bar by selecting the services tab. Upon entering the treatment and services page, user is given a list of available treatment which user can interact with by clicking on the hyperlink which is the name of the treatment itself. Upon interacting with the proposed element, users will be redirected into the treatment details page where a detailed explanation on the particular

treatment are displayed to the user. In the webpage itself, user is given notice on the consistency of the header and footer which can be seen across multiple webpages along with the main content, which is the treatment details. In the main body of the webpage lies the treatment details, which provide insight on the selected treatment previously chosen by the user. For example, a description on the treatment is shown to the user as well as possible self-checking tips which can be utilized by the user before having to visit the clinic themselves for a check-up. Aside from that, the user will also be given any possible suggestion on treating the possible condition that the user might be facing as well as the available doctors in the clinic which would provide consultation based on the particular medical condition.

## 15.1.9 Our Doctor

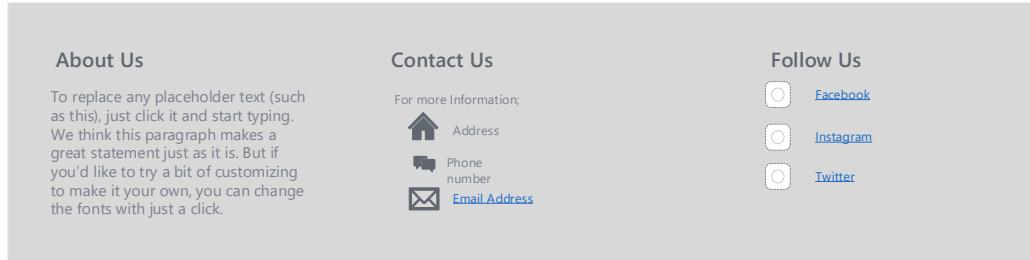
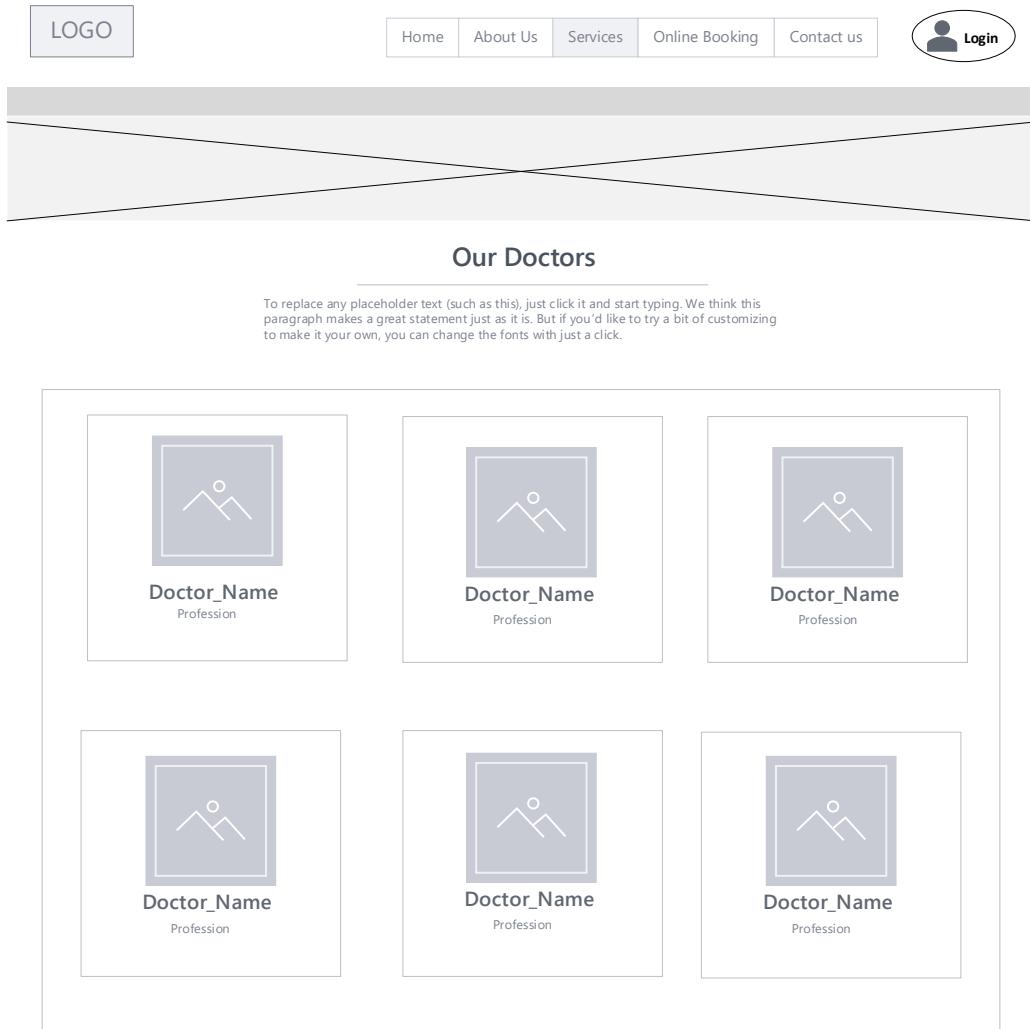
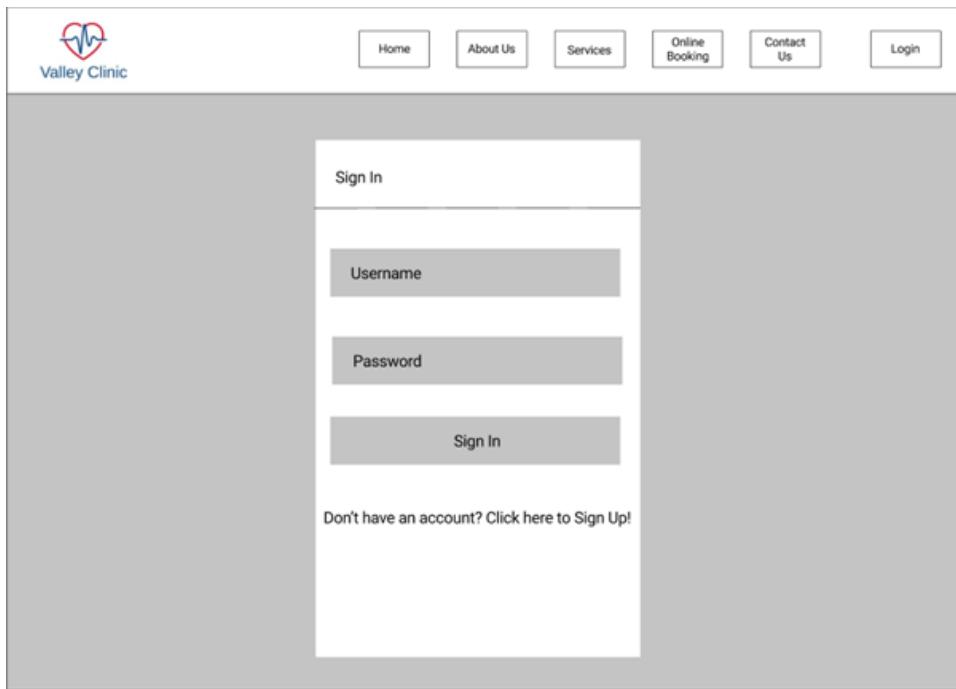


Figure 22 Valley Clinic Our Doctors webpage

Based on Figure 20 which depicts the layout of Doctor Details Information page, starting of the webpage with the consistent header which can be seen through the navigation bar and login button. Aside from that, the content of which as the title shown are information on the available

doctors on duty in Valley Clinic. Upon visiting the particular webpage, users will be given an insight on who they are to expect to see during their treatment session and the particular specialist of their field. Aside from that, a brief description on the work ethics is also shown under “Our Doctors” heading.

### 15.1.10 Sign In

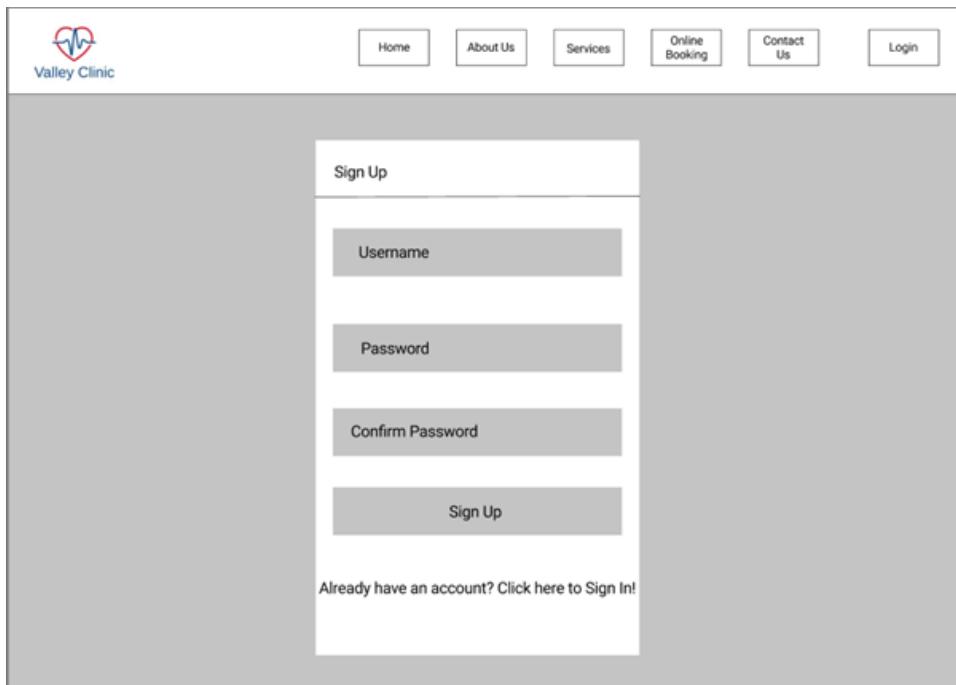


The screenshot shows the sign-in page for Valley Clinic. At the top left is the Valley Clinic logo, which includes a stylized heart icon. To the right of the logo is a horizontal navigation bar with five buttons: "Home", "About Us", "Services", "Online Booking", and "Contact Us". On the far right of the navigation bar is a "Login" button. The main content area has a light gray background. It features a white rectangular form with a thin black border. The form has a header section labeled "Sign In". Below this are two input fields: the top one is labeled "Username" and the bottom one is labeled "Password", both in a small, dark font. At the bottom of the form is a large, dark gray button labeled "Sign In" in white. Below the form, there is a small, centered text link: "Don't have an account? Click here to Sign Up!".

Figure 23 Valley Clinic Sign In page

Based on the picture above, if the user is registered on our website they can just sign-in to their own accounts by clicking on the login button and they will be redirected to this page and the user can enter their own username and password to access into their own accounts.

### 15.1.11 Sign Up

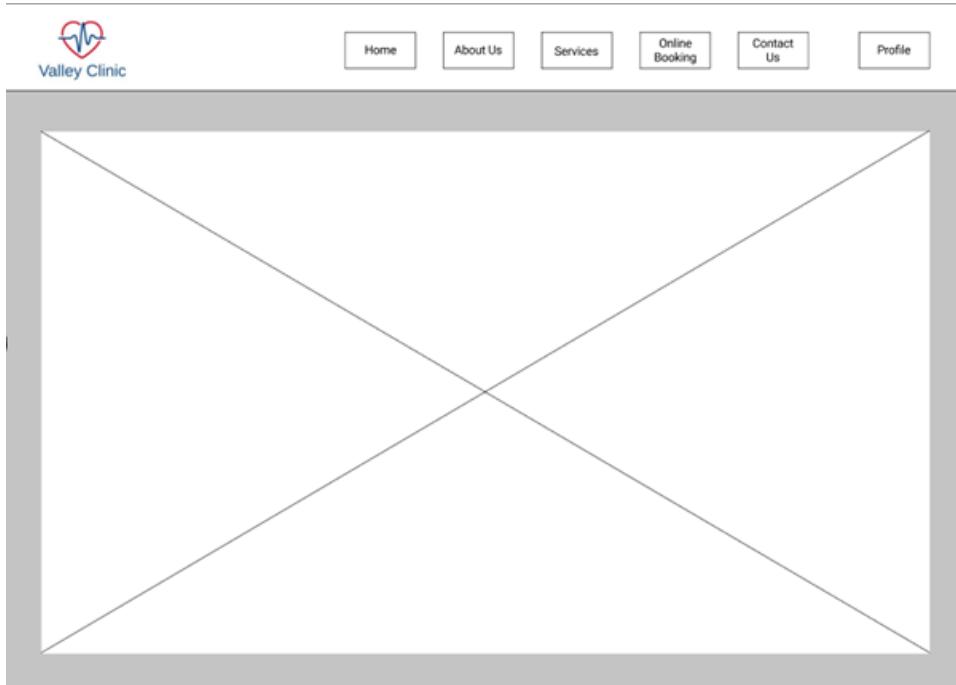


The screenshot shows the 'Sign Up' page of the Valley Clinic website. At the top, there is a navigation bar with links for Home, About Us, Services, Online Booking, Contact Us, and Login. The 'Login' link is highlighted with a red box. Below the navigation bar, the page title 'Sign Up' is displayed. The form consists of four input fields: 'Username', 'Password', 'Confirm Password', and a 'Sign Up' button. Below the form, a link 'Already have an account? Click here to Sign In!' is visible.

Figure 24 Valley Clinic Sign Up page

Based on the picture above, when the user decides to create an account, they will need to click on the login button and select “Sign Up”. Once they’ve clicked the button, the user will be redirected to this page where the user will key in their own username, password and enter their own password one more time as confirmation. If the fields are empty, the system will issue an error message “Fields cannot be empty!”. Therefore, data validation will be present.

### 15.1.12 Redirect User Index



*Figure 25 Valley Clinic Customer webpage*

Once the user has successfully logged in, they will be redirected to this page where they can choose to view what the clinic offers.

### 15.1.13 User Profile

The screenshot shows a web page titled "Member Profile". At the top, there is a navigation bar with links for Home, About Us, Services, Online Booking, Contact Us, and Profile. The "Profile" link is highlighted. On the left, there is a logo for "Valley Clinic" featuring a heart icon. The main content area is a form with five input fields: "Username", "Password", "Email", "Phone number", and "Address". Below the form are two buttons: "Edit" on the left and "Save" on the right.

Figure 26 Valley Clinic Customer Profile webpage

If the user wants to make changes to their account, they will need to click on their profile. They will be redirected to this page where it will display their current details of their accounts. If the user wishes to make changes to their account, they will need to click on the edit button and once they have confirmed the changes the user will need to click on the save button.

### 15.1.13 Book Appointment

The screenshot shows a webpage for 'Valley Clinic' with a navigation bar at the top featuring links for Home, About Us, Services, Online Booking, Contact Us, and Profile. Below the navigation is a table titled 'Customer Bookings'. The table has four columns: 'No.', 'Information', 'Date', and 'Options'. It contains two rows of data:

No.	Information	Date	Options
1	X000XXXX	14/2/2022	<button>Book</button>
2	X000XXXX	1/3/2022	<button>Book</button>

Figure 27 Valley Clinic Customer Bookings webpage

Based on the picture above, the user can choose appointments to be booked. If the treatment and date suites their criteria, they will need to click on the book button to proceed with the confirmation with their booking.

## 15.1.14 Payment



Valley Clinic

Home    About Us    Services    Online Booking    Contact Us    Login

Select Payment

Cash

Card



Valley Clinic

Home    About Us    Services    Online Booking    Contact Us    Login

Payment

Name

Card Number

CVV

Date

Submit

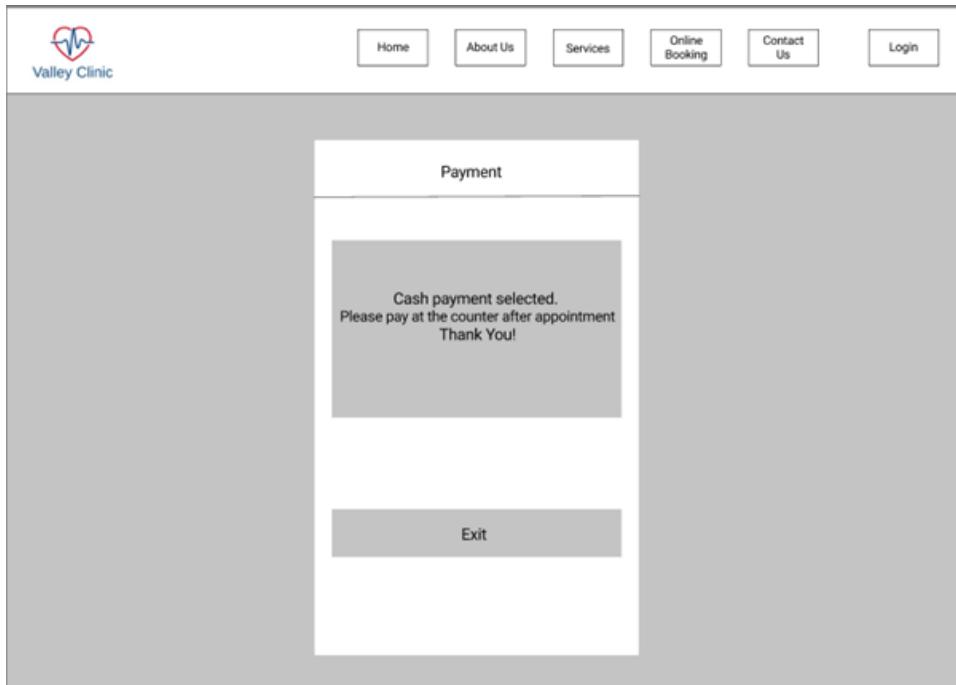
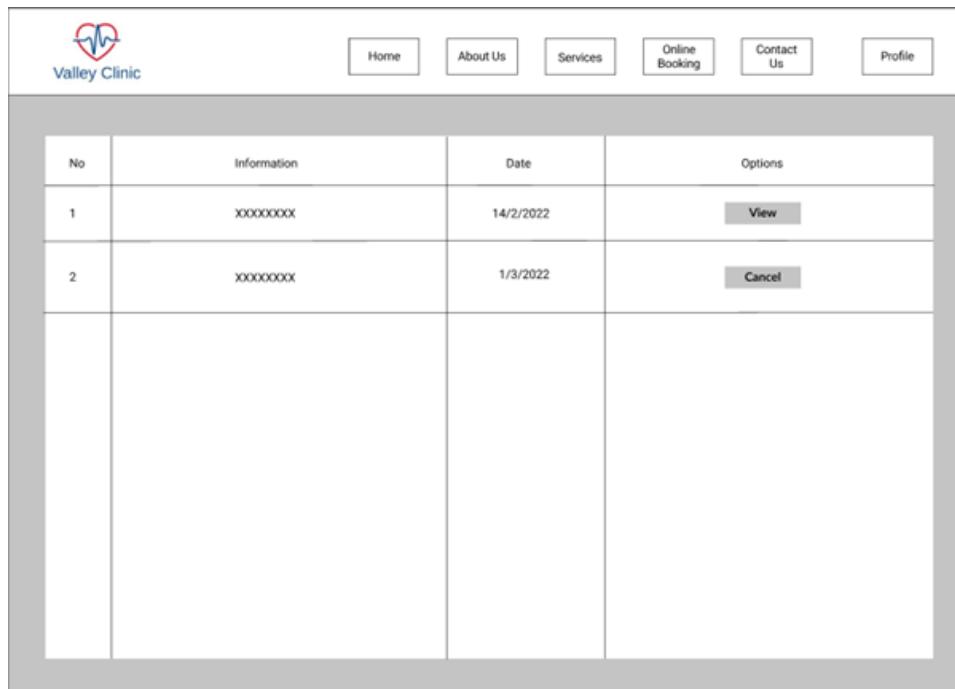


Figure 28 Valley Clinic Payment webpage

Based on the first picture, once the user has clicked on the book button on the appointments page they will be redirected to a page where they can choose between Card or Cash as their payment. If the user chooses Card, they will need to enter their details such as their name, card number, cvv and date based on the 2<sup>nd</sup> picture. If the user chooses to pay with cash, a message will be displayed to the user stating “Cash payment selected. Please pay at the counter after the appointment. Thank You!”.

### 15.1.15 Booking History



The screenshot shows a web page for 'Valley Clinic'. At the top left is the clinic's logo, which includes a stylized heart icon and the text 'Valley Clinic'. To the right of the logo is a horizontal menu bar with five items: 'Home', 'About Us', 'Services', 'Online Booking', 'Contact Us', and 'Profile'. Below the menu is a table titled 'Booking History'. The table has four columns: 'No.', 'Information', 'Date', and 'Options'. There are two rows of data in the table. Row 1 contains information for appointment number 1, with date 14/2/2022 and an 'Options' column containing a single 'View' button. Row 2 contains information for appointment number 2, with date 1/3/2022 and an 'Options' column containing a single 'Cancel' button.

No.	Information	Date	Options
1	XXXXXXX	14/2/2022	<a href="#">View</a>
2	XXXXXXX	1/3/2022	<a href="#">Cancel</a>

Figure 29 Valley Clinic Customer Booking History webpage

Based on the picture above, the user can view their booking history that they've made with our system. The user can click on the view button to view the appointment that they have already attended. If the user wants to cancel their appointment, they will be able to cancel before the specified date.

### 15.1.15 Booking Invoice

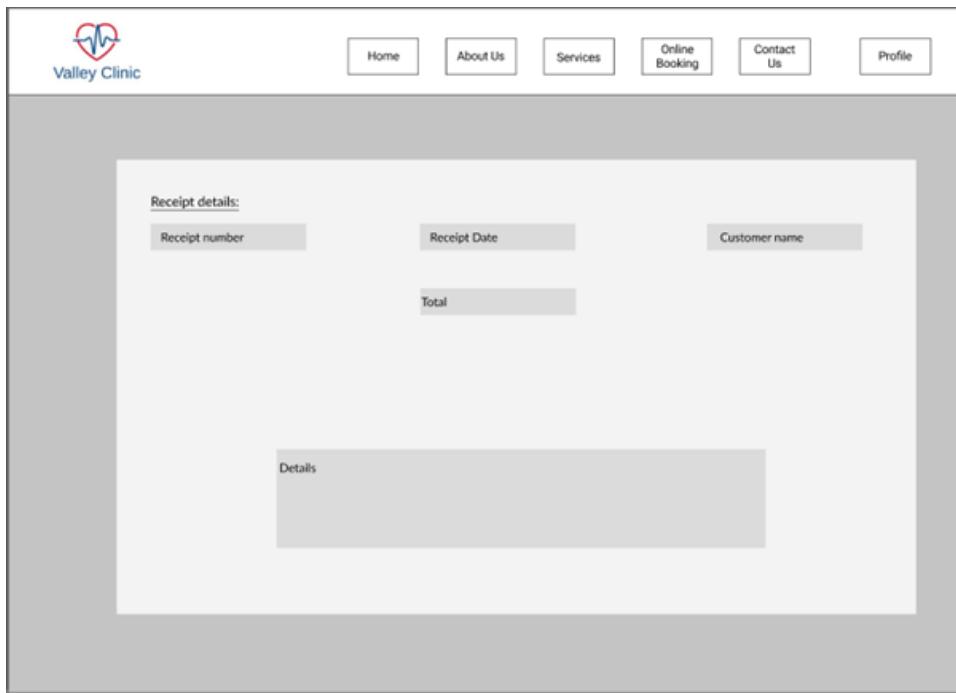


Figure 30 Valley Clinic Customer receipt webpage

Once the user has clicked on the view button on the booking history page, a receipt will be generated for them to see the information of the appointment.

## 15.2 Administrator Module

### 15.2.1 Login

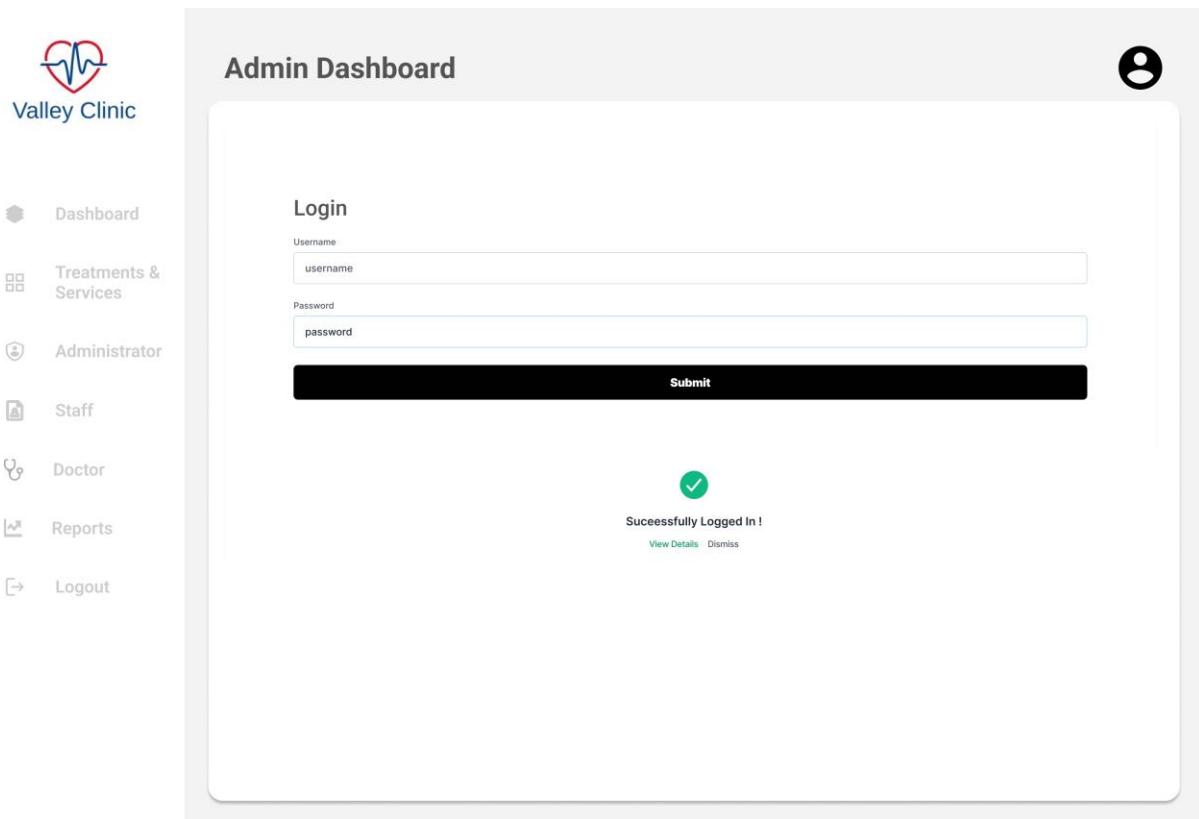


Figure 23: Admin Dashboard Login Wireframe

To access the admin dashboard, admin need to login with their username and password in this login page.

## 15.2.2 Dashboard

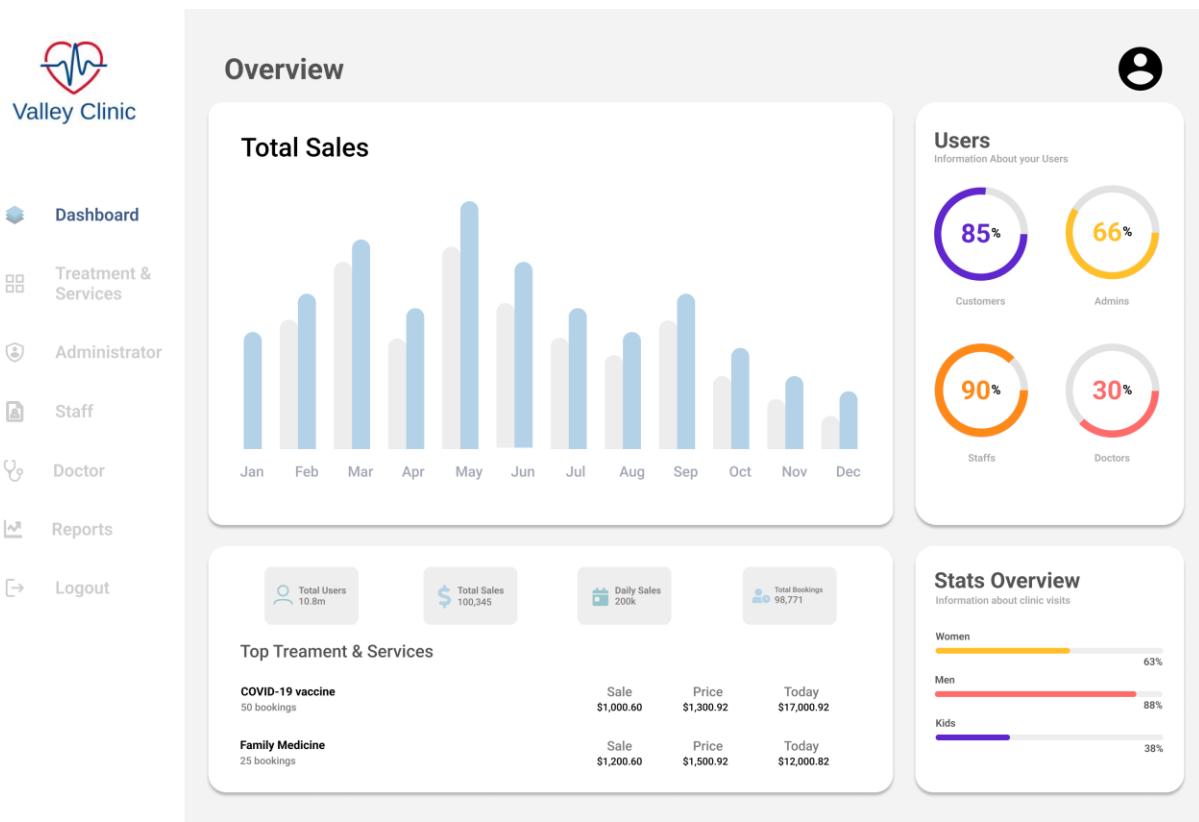
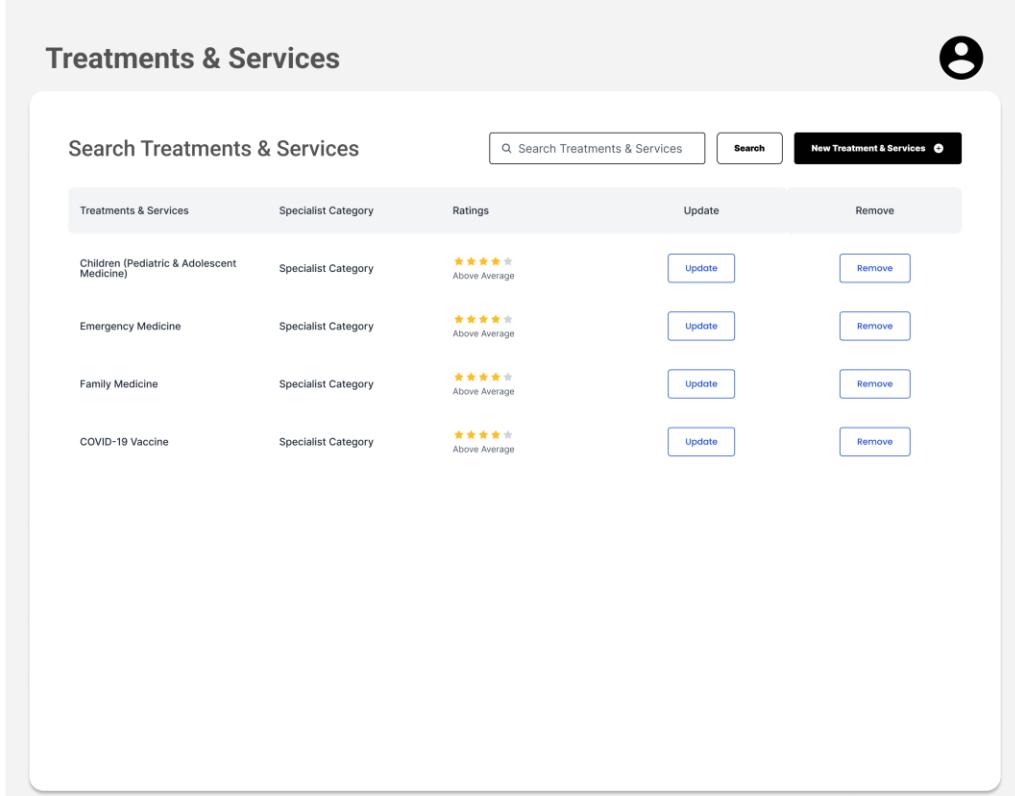


Figure 24: Admin Dashboard Overview Wireframe

After admin successfully logged in, admin will be redirected to this dashboard overview page. In this page, there are a bar chart showing the total sales, a circular progress bar showing percentage of each type of users, a section which show some information and stats overview that shows information of clinic visitors.

### 15.2.3 Manage Treatment & Services

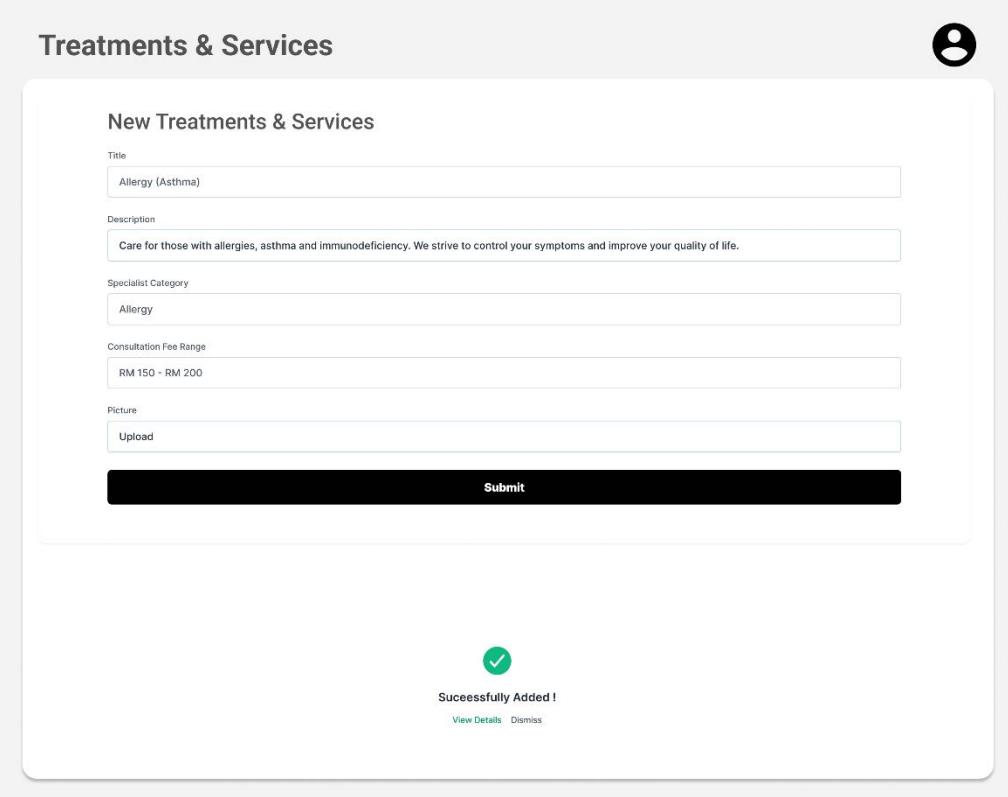


The wireframe shows the Admin Dashboard's Treatment & Services page. On the left, a sidebar menu includes: Dashboard, Treatments & Services (selected), Administrator, Staff, Doctor, Reports, and Logout. The main content area has a header 'Treatments & Services' with a search bar and a 'New Treatment & Services' button. Below is a table with columns: Treatment & Services, Specialist Category, Ratings, Update, and Remove. Data rows include: Children (Pediatric & Adolescent Medicine) with 4.5 stars (Above Average), Emergency Medicine with 4.5 stars (Above Average), Family Medicine with 4.5 stars (Above Average), and COVID-19 Vaccine with 4.5 stars (Above Average). Each row has 'Update' and 'Remove' buttons.

Treatments & Services	Specialist Category	Ratings	Update	Remove
Children (Pediatric & Adolescent Medicine)	Specialist Category	★★★★★ Above Average	Update	Remove
Emergency Medicine	Specialist Category	★★★★★ Above Average	Update	Remove
Family Medicine	Specialist Category	★★★★★ Above Average	Update	Remove
COVID-19 Vaccine	Specialist Category	★★★★★ Above Average	Update	Remove

Figure 25: Admin Dashboard Treatment & Services Wireframe

To view all the treatment and services, admin can navigate to the menu bar on the left panel and click on the “Treatment & Services”. In this page, admin gets to search for the treatment, and view the specialist category and ratings. Furthermore, admin can also update or remove the treatment and services by clicking on the respective button. Other than that, admin can also click on the row of treatment and services to view all the details. Lastly, there will be a “New Treatment & Services” button for the admin to insert new treatment and services into the system.



The wireframe shows the 'Treatments & Services' section of the admin dashboard. On the left, a sidebar lists navigation options: Dashboard, Treatments & Services (selected), Administrator, Staff, Doctor, Reports, and Logout. The main content area is titled 'Treatments & Services' and contains a 'New Treatments & Services' form. The form fields include: Title (Allergy (Asthma)), Description (Care for those with allergies, asthma and immunodeficiency. We strive to control your symptoms and improve your quality of life.), Specialist Category (Allergy), Consultation Fee Range (RM 150 - RM 200), and Picture (Upload). A 'Submit' button is at the bottom. A success message 'Successfully Added!' with a checkmark icon and links to 'View Details' and 'Dismiss' is displayed at the bottom right.

**Treatments & Services**

New Treatments & Services

Title  
Allergy (Asthma)

Description  
Care for those with allergies, asthma and immunodeficiency. We strive to control your symptoms and improve your quality of life.

Specialist Category  
Allergy

Consultation Fee Range  
RM 150 - RM 200

Picture  
Upload

Submit

Successfully Added!

[View Details](#) [Dismiss](#)

Figure 26: Admin Dashboard New Treatment & Services Wireframe

To insert a new treatment and services, admin needs to navigate to the top right corner of the treatment and services page and click on the button. After that, admin can fill up this form and submit. The new treatment and services will be updated.



- ➊ Dashboard
- ➋ Treatments & Services
- ➌ Administrator
- ➍ Staff
- ➎ Doctor
- ➏ Reports
- ➐ Logout

## Treatments & Services

➊

**Update Treatments & Services**

**Title**  
Allergy (Asthma)

**Description**  
Care for those with allergies, asthma and immunodeficiency. We strive to control your symptoms and improve your quality of life.

**Specialist Category**  
Allergy

**Consultation Fee Range**  
RM150 - RM 200

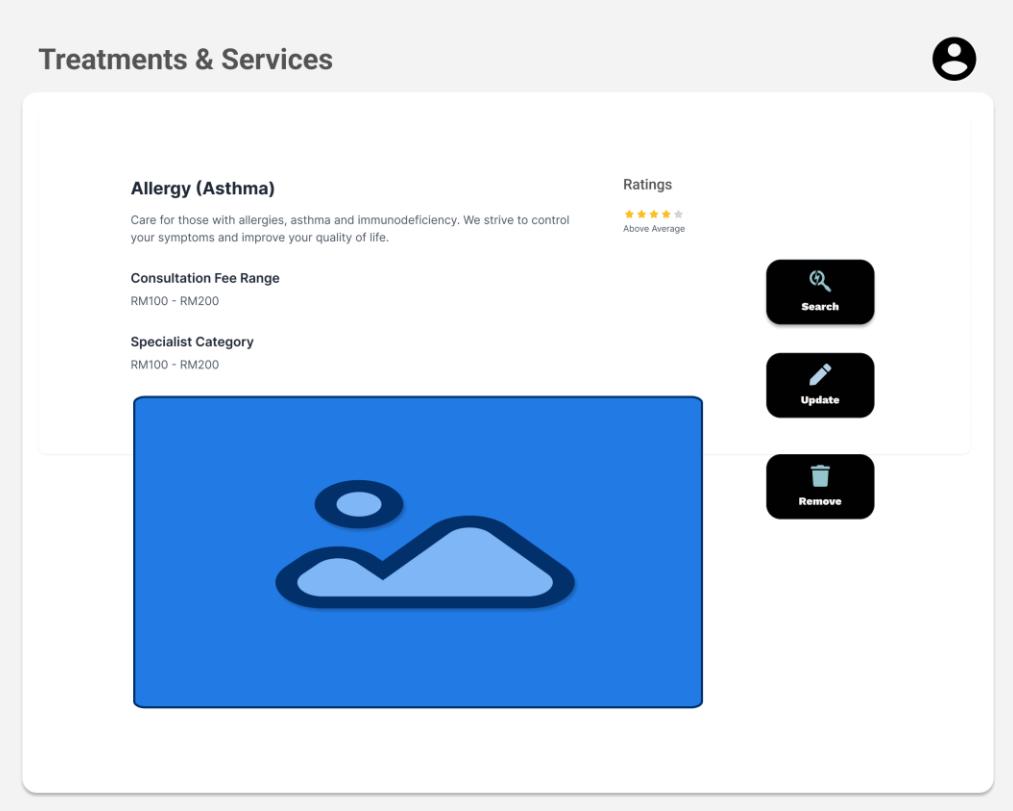
**Picture**  
Upload

**Submit**

✓  
**Successfully Updated !**  
[View Details](#) [Dismiss](#)

Figure 27: Admin Dashboard Update Treatment & Services Wireframe

To update treatment and services, admin can click on the “Update” button on that row of treatment and services. After admin done updating the information, the treatment and services will be updated accordingly.



The wireframe shows the Admin Dashboard interface. On the left is a sidebar with a logo (a heart with a red line) and the text "Valley Clinic". The sidebar contains the following menu items:

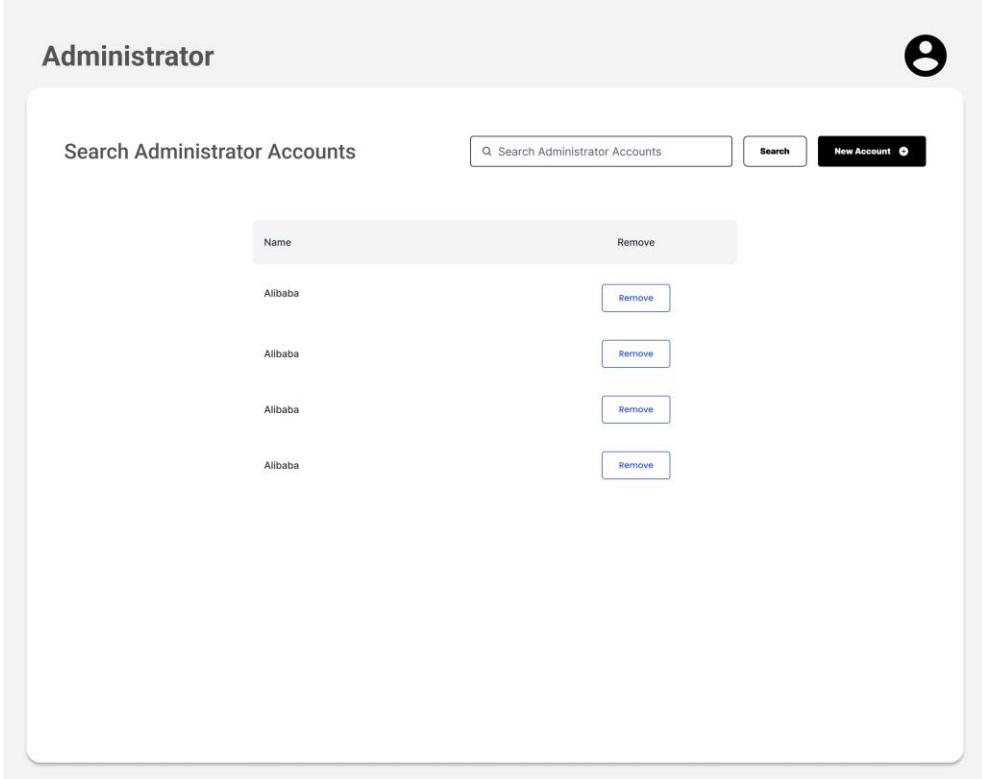
- Dashboard
- Treatments & Services** (highlighted with a blue square)
- Administrator
- Staff
- Doctor
- Reports
- Logout

The main content area is titled "Treatments & Services". It displays a row for "Allergy (Asthma)". The row includes the service name, a brief description ("Care for those with allergies, asthma and immunodeficiency. We strive to control your symptoms and improve your quality of life."), a "Ratings" section with a 5-star icon and the text "Above Average", and three action buttons: "Search", "Update", and "Remove". A large blue placeholder image of a cloud with a sun is centered below the row.

Figure 28: Admin Dashboard View Treatment & Services Wireframe

Admin can view all the details of a particular treatment and services after they click on the row of treatment and services. There will also be search, update, and remove functions provided to the admin.

#### 15.2.4 Manage Administrator

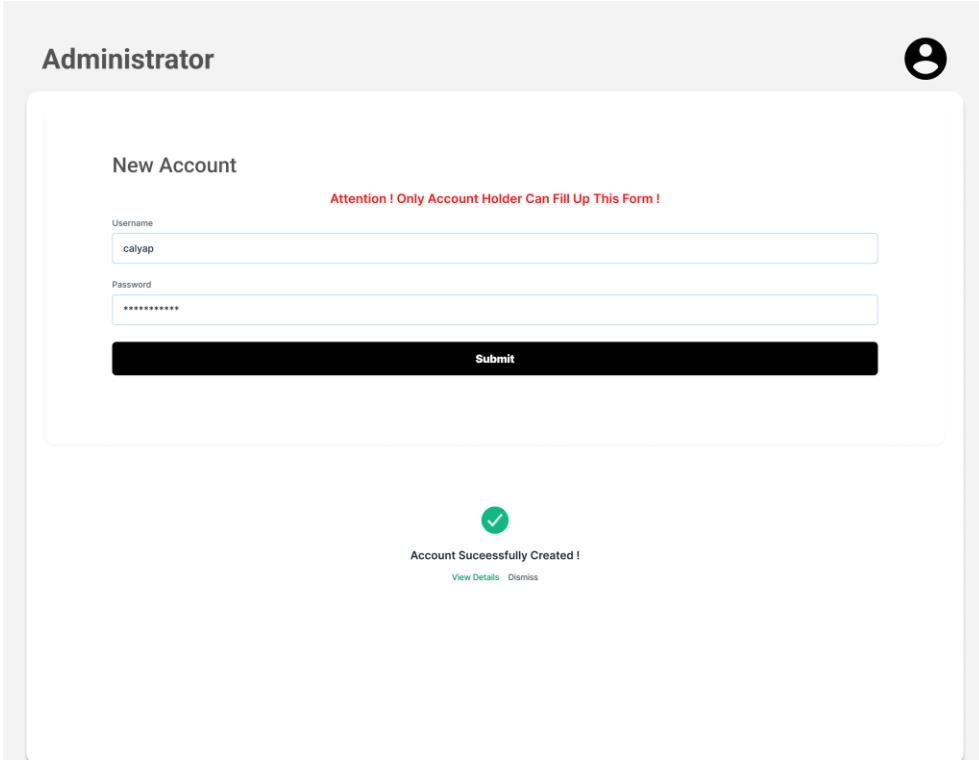


The wireframe shows the 'Administrator' section of the admin dashboard. On the left, there's a sidebar with a 'Valley Clinic' logo and a heart icon. The sidebar menu includes: Dashboard, Treatments & Services, Administrator (selected), Staff, Doctor, Reports, and Logout. The main area is titled 'Administrator' and contains a search bar with placeholder 'Search Administrator Accounts' and a 'Search' button. Below the search bar is a table with a single row showing an account named 'Alibaba'. The table has columns for 'Name' and 'Remove'. There are three 'Remove' buttons, one for each account entry.

Name	Remove
Alibaba	Remove

Figure 29: Admin Dashboard Search Administrator Account Wireframe

Admin can navigate to this page from the menu bar on the left panel. In this page, admin gets to search through all the admin accounts and can remove the accounts. To create a new admin account, a “New Account” button is provided.

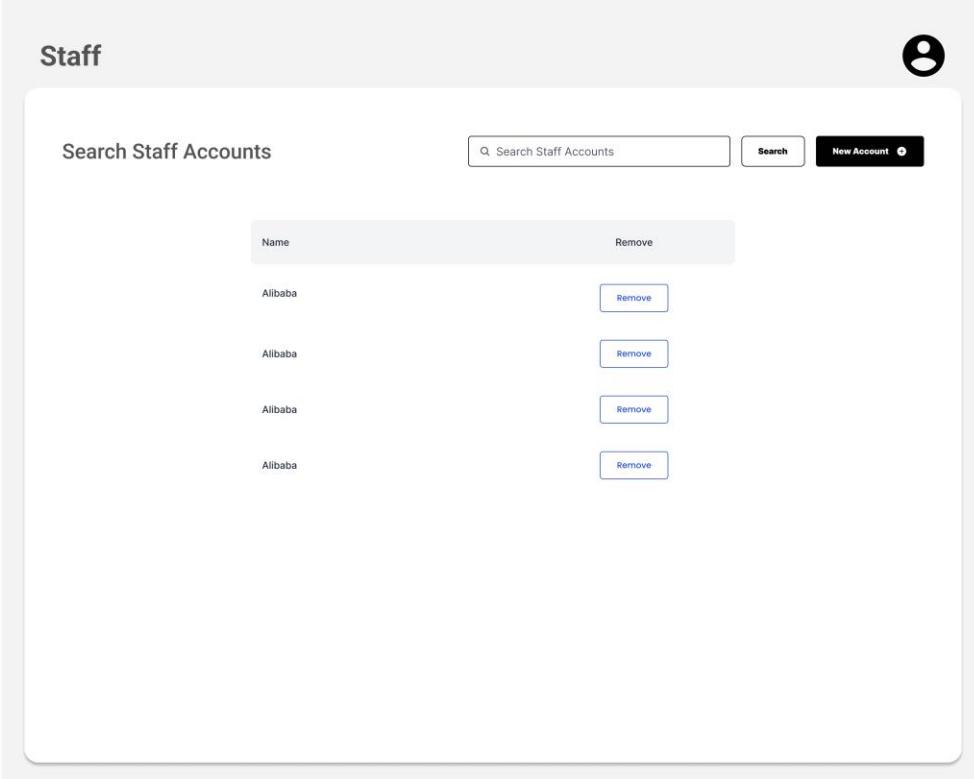


The wireframe shows the 'Administrator' dashboard of the Valley Clinic system. On the left, a sidebar lists navigation options: Dashboard, Treatments & Services, Administrator (selected), Staff, Doctor, Reports, and Logout. The main content area is titled 'Administrator' and contains a 'New Account' form. The form includes fields for 'Username' (containing 'calyp') and 'Password' (containing '\*\*\*\*\*'). A red warning message at the top right reads 'Attention ! Only Account Holder Can Fill Up This Form !'. A large black button labeled 'Submit' is centered below the form. At the bottom right, a green checkmark icon indicates 'Account Successfully Created !' with links to 'View Details' and 'Dismiss'.

Figure 30: Admin Dashboard New Administrator Account Wireframe

After admin click on the “New Account” button, this page will be displayed. To avoid other anonymous user simply register an account in the system, admin need assistance from another administrator to create a new account. The username and password can only be input by the account holder.

### 15.2.5 Manage Staff



The wireframe shows the 'Staff' section of the admin dashboard. At the top left is the Valley Clinic logo. On the right is a user icon. Below the logo is a sidebar menu with the following items:

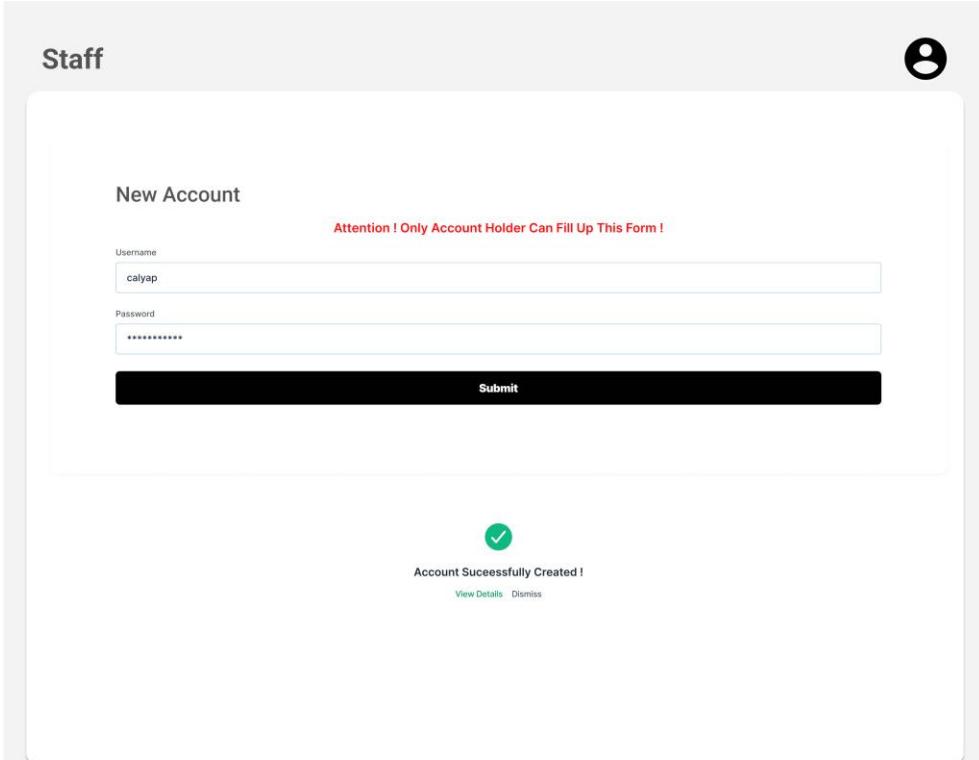
- Dashboard
- Treatments & Services
- Administrator
- Staff** (highlighted)
- Doctor
- Reports
- Logout

The main content area is titled 'Staff' and contains a search bar with placeholder 'Search Staff Accounts' and a 'Search' button. Below the search bar is a table with four rows, each showing a staff account named 'Alibaba' and a 'Remove' button to its right.

Name	Remove
Alibaba	Remove
Alibaba	Remove
Alibaba	Remove

Figure 31: Admin Dashboard Search Staff Account Wireframe

Admin can navigate to this page from the menu bar on the left panel. In this page, admin gets to search through all the staff accounts and can remove the accounts. To create a new staff account, a “New Account” button is provided.

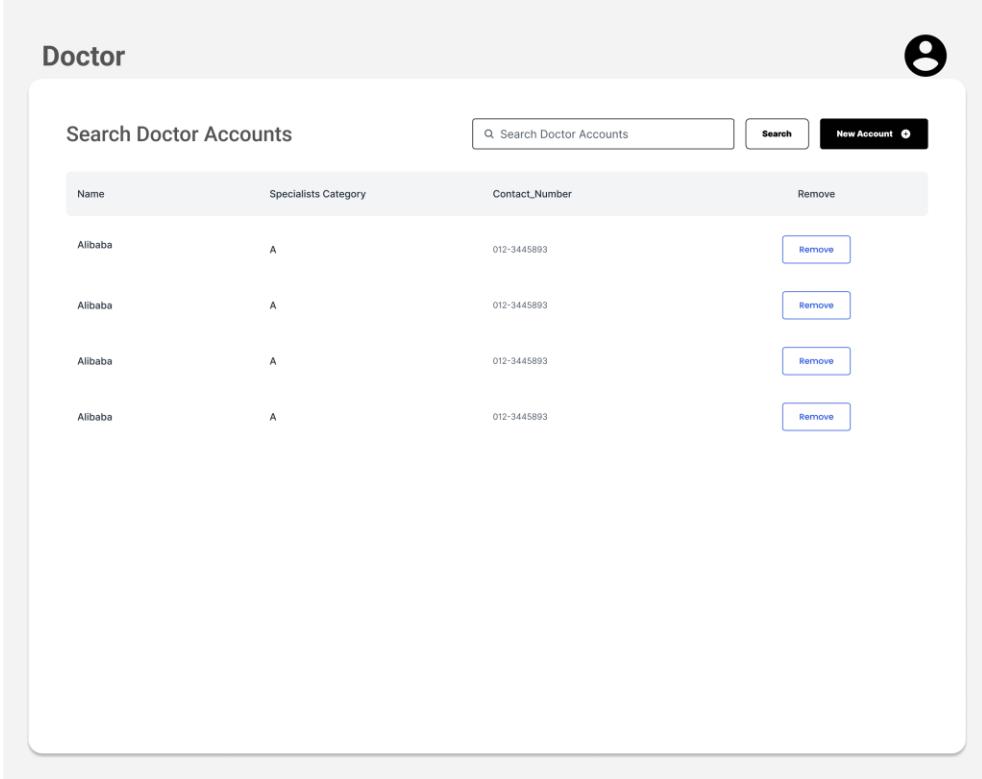


The wireframe shows the 'Staff' section of the admin dashboard. On the left, a sidebar lists navigation options: Dashboard, Treatments & Services, Administrator, Staff (selected), Doctor, Reports, and Logout. The main content area is titled 'New Account' and contains a note: 'Attention ! Only Account Holder Can Fill Up This Form !'. It features two input fields: 'Username' (containing 'calyap') and 'Password' (containing '\*\*\*\*\*'). A large black 'Submit' button is centered below the fields. At the bottom right, a green circular icon with a checkmark is followed by the message 'Account Suceessfully Created !' and links to 'View Details' and 'Dismiss'.

Figure 32: Admin Dashboard New Staff Account Wireframe

After admin click on the “New Account” button, this page will be displayed. To avoid other anonymous user simply register an account in the system, staff need assistance from the administrator to create a new account. The username and password can only be input by the account holder.

### 15.2.6 Manage Doctor

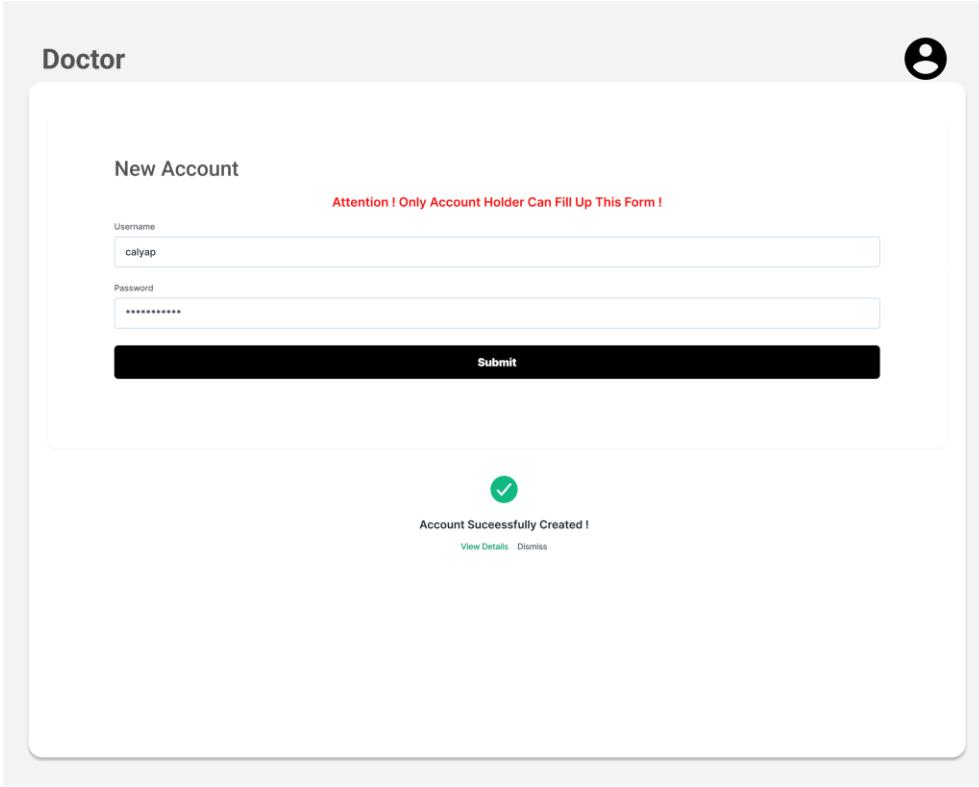


The wireframe shows the 'Doctor' page of the admin dashboard. At the top left is the 'Valley Clinic' logo with a heart icon. On the right is a user profile icon. The main title 'Doctor' is centered above a search bar with placeholder 'Search Doctor Accounts' and a 'Search' button. To the right of the search bar is a 'New Account' button with a plus sign. Below the search area is a table with four columns: 'Name', 'Specialists Category', 'Contact\_Number', and 'Remove'. There are four rows of data, each showing 'Alibaba' in the Name column, 'A' in the Specialists Category column, '012-3445893' in the Contact\_Number column, and a 'Remove' button in the fourth column.

Name	Specialists Category	Contact_Number	Remove
Alibaba	A	012-3445893	<button>Remove</button>
Alibaba	A	012-3445893	<button>Remove</button>
Alibaba	A	012-3445893	<button>Remove</button>
Alibaba	A	012-3445893	<button>Remove</button>

Figure 33: Admin Dashboard Search Doctor Account Wireframe

To view all the doctor accounts, admin can navigate to the menu bar on the left panel and click on the “Doctor”. In this page, admin gets to search for the doctor, and view the specialist category and contact number. Furthermore, admin can also remove the doctor account by clicking on the respective button. Other than that, admin can also click on the row of doctor to view all the details. Lastly, there will be a “New Account” button for the admin to insert new doctor into the system.



The wireframe shows the 'New Account' form for a doctor. It includes fields for 'Username' (containing 'calyap') and 'Password' (containing '\*\*\*\*\*'). A 'Submit' button is at the bottom. A success message 'Account Successfully Created!' with a checkmark icon is displayed below the form.

Valley Clinic

Doctor

New Account

Attention ! Only Account Holder Can Fill Up This Form !

Username  
calyap

Password  
\*\*\*\*\*

Submit

✓

Account Successfully Created !

[View Details](#) [Dismiss](#)

Dashboard

Treatments & Services

Administrator

Staff

Doctor

Reports

Logout

Figure 34: Admin Dashboard New Doctor Account Wireframe

After admin click on the “New Account” button, this page will be displayed. To avoid other anonymous user simply register an account in the system, doctor need assistance from the administrator to create a new account. The username and password can only be input by the account holder.

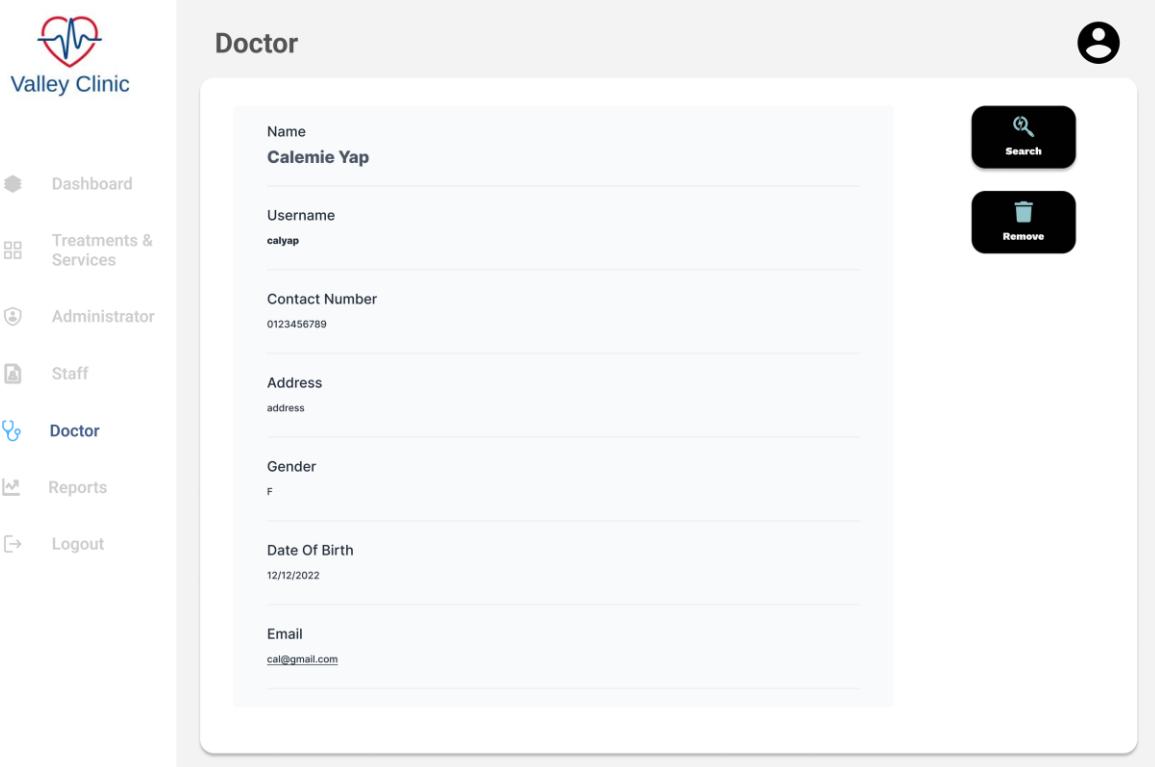


Figure 35: Admin Dashboard View Doctor Account Wireframe

Admin can view all the details of a particular doctor after admin click on the row of doctor account. There will also be search and remove functions provided to the admin.

## 15.2.7 Reports

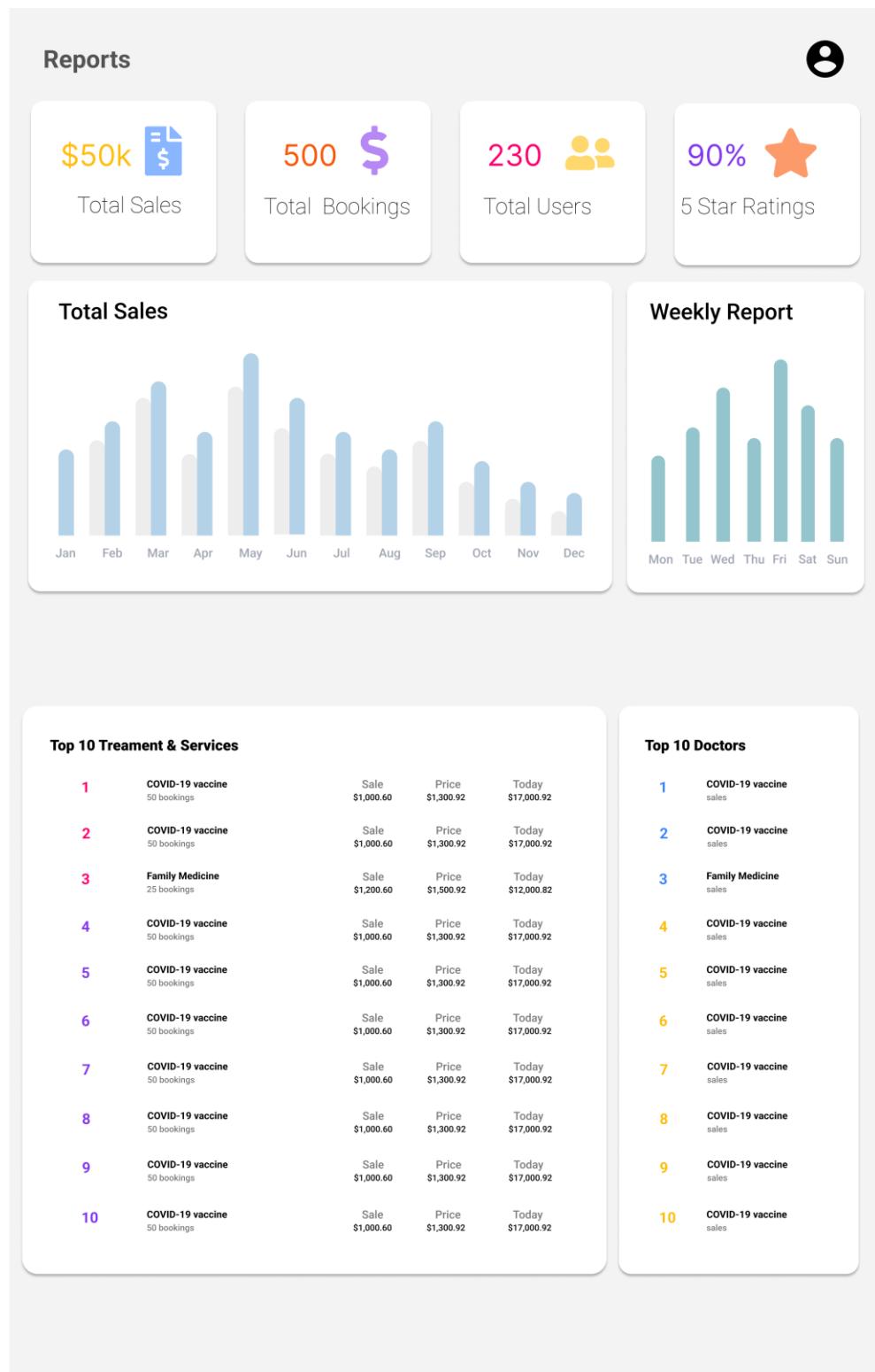


Figure 36: Admin Dashboard Report Wireframe

To view the reports, admin can navigate to the menu bar on the left panel and click on the “Reports”. In this page, admin gets to view the total sales, total bookings, total users, and percentage of 5-star ratings. Furthermore, there will be bar charts of total sales and weekly sales. Lastly, the top 10 treatment and services and doctors will be displayed along with details.

### 15.2.8 Logout

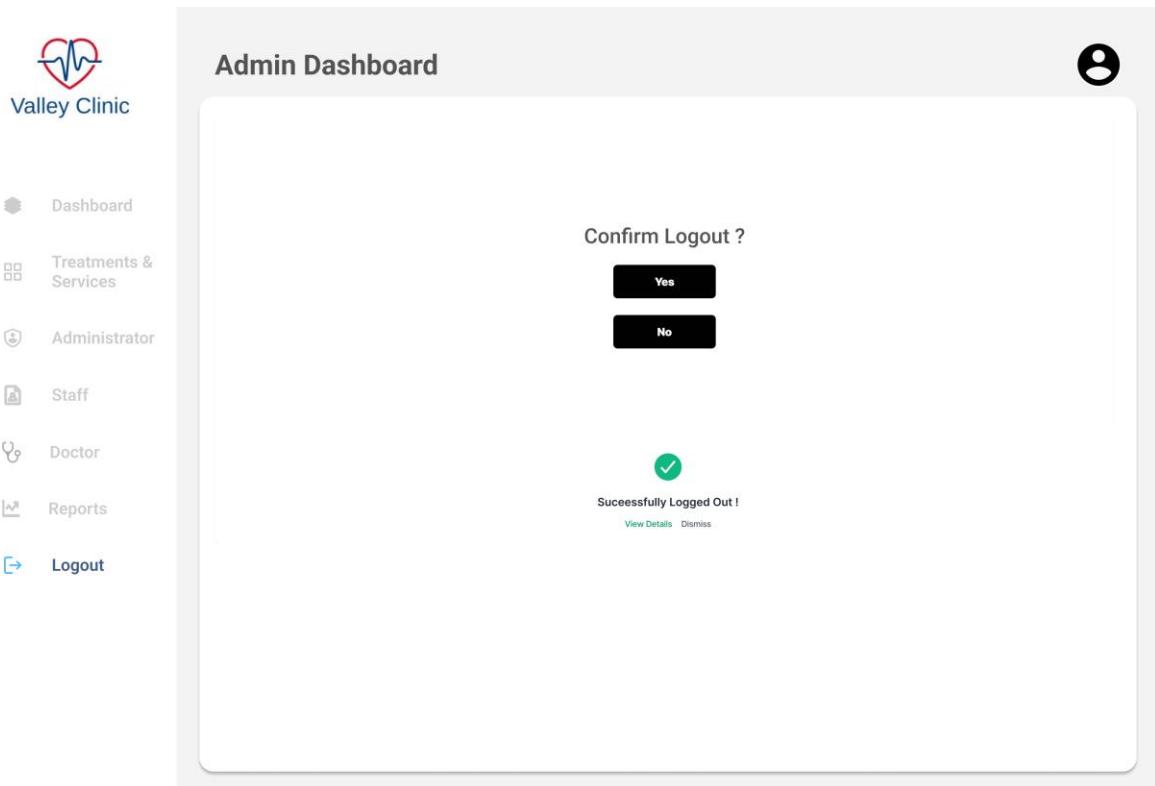
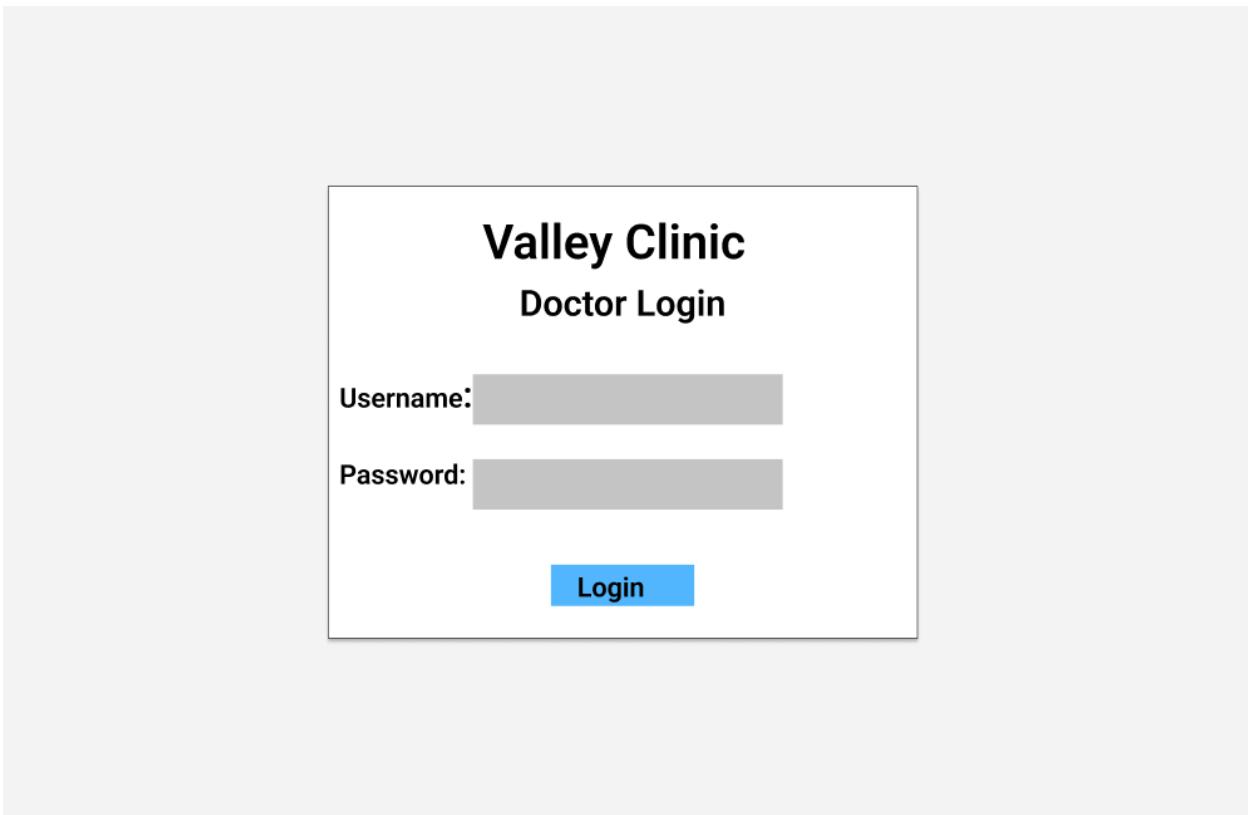


Figure 37: Admin Dashboard Logout Wireframe

This is the logout page where admin can login by clicking on the “Logout” at the menu bar. If admin click “Yes” admin will logged out, else admin will go back to the previous page.

## 15.3 Doctor Module

### 15.3.1 Doctor Login



A wireframe diagram of a doctor login page. The page has a light gray background. In the center, there is a white rectangular form. At the top of the form, the text "Valley Clinic" is displayed in bold black font, followed by "Doctor Login" in a slightly smaller bold black font. Below this, there are two input fields: one labeled "Username:" and another labeled "Password:", both preceded by their respective labels in bold black font. At the bottom of the form is a blue rectangular button with the word "Login" in white.

*Figure 38 Doctor Login Wireframe*

This is the doctor login page of Valley Clinic; the doctor will need to enter his/her username and password to login to his/her personal account.

### 15.3.2 Doctor Homepage

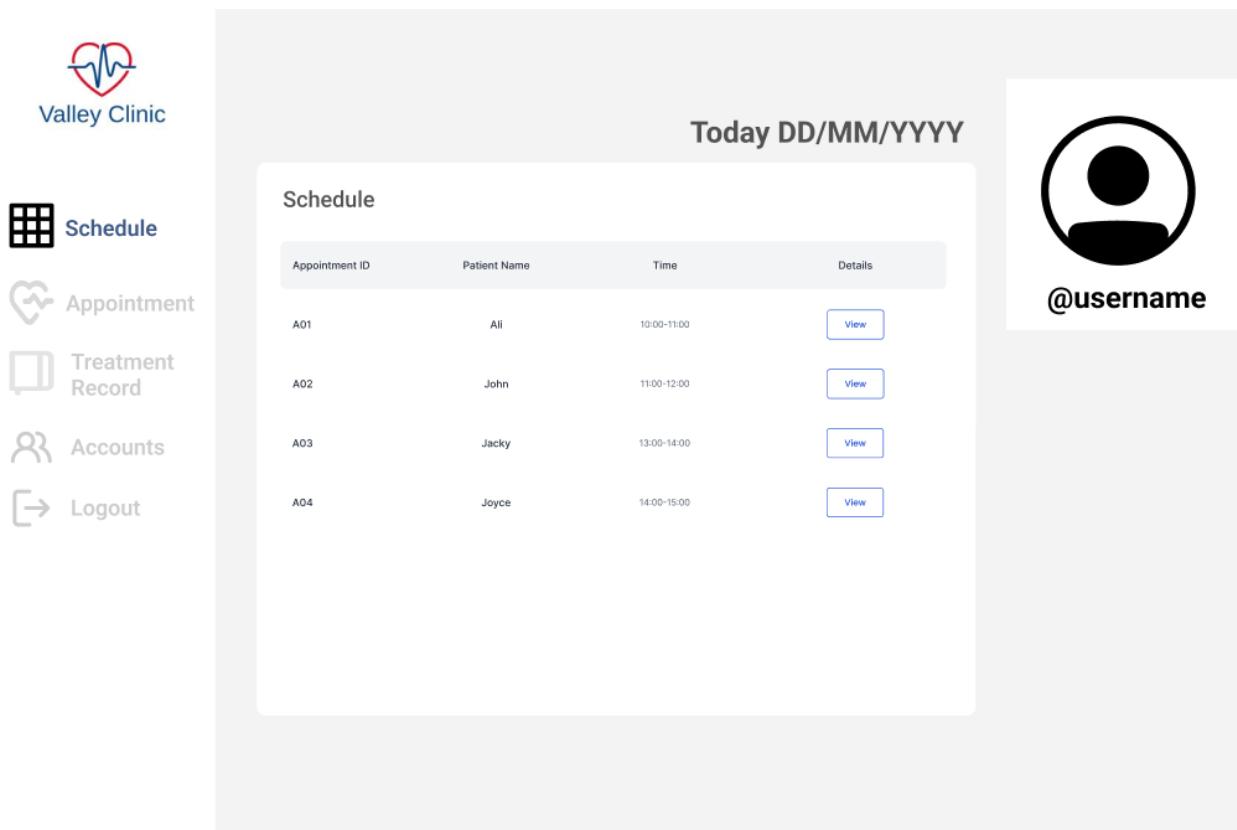


Figure 39 Doctor Homepage Wireframe

This is the homepage of the doctor website, where the users will be directed to this page after successfully logging in to their account. As shown in the figure, in this homepage it will show information about daily schedule of the doctor and on the right-hand side it will appear doctor profile picture and username.

### 15.3.2 Doctor Appointment Page

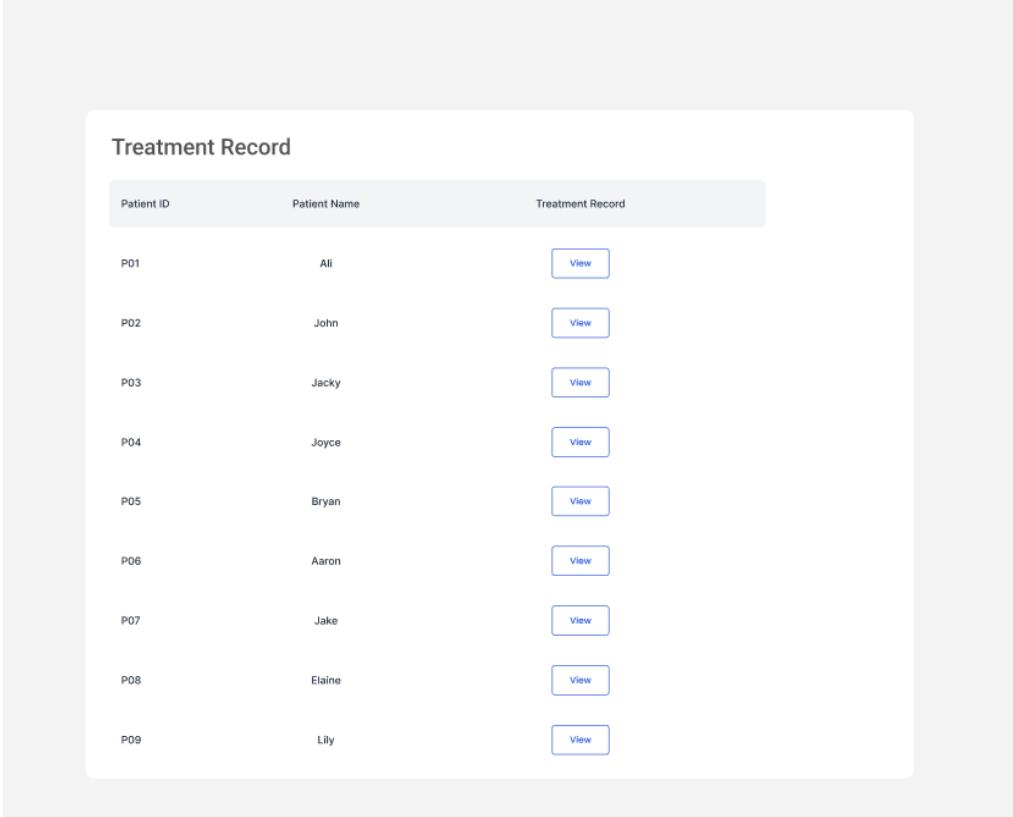
The wireframe illustrates the Doctor Appointment Page. On the left, a vertical navigation bar for 'Valley Clinic' features icons and labels for 'Schedule', 'Appointment', 'Treatment Record', 'Accounts', and 'Logout'. The main content area is titled 'Appointment' and includes a search bar ('Search by Appointment ID, Patient Name') and a 'Filter' button. A data table lists nine appointments with columns for 'Appointment ID', 'Patient Name', 'Time', and 'Date'.

Appointment ID	Patient Name	Time	Date
A01	Ali	10:00-11:00	14/2/2022
A02	John	11:00-12:00	14/2/2022
A03	Jacky	13:00-14:00	14/2/2022
A04	Joyce	14:00-15:00	14/2/2022
A05	Bryan	10:00-11:00	15/2/2022
A06	Adrian	10:00-11:00	15/2/2022
A07	Jake	11:00-12:00	15/2/2022
A08	Lily	13:00-14:00	15/2/2022
A09	Elaine	14:00-15:00	15/2/2022

Figure 40 Doctor Appointment Page Wireframe

This is the appointment page which show all information about the appointment. User can direct to this page by clicking on the button of the left side navigation bar. User can also search for their desire information by inserting keywords on the input bar above the data table.

### 15.3.3 Doctor Treatment Record Page



The wireframe for the Treatment Record page shows a sidebar with a logo and navigation links, and a main content area with a table of patient records.

**Valley Clinic**

Schedule

Appointment

Treatment Record

Accounts

Logout

**Treatment Record**

Patient ID	Patient Name	Treatment Record
P01	Ali	<a href="#">View</a>
P02	John	<a href="#">View</a>
P03	Jacky	<a href="#">View</a>
P04	Joyce	<a href="#">View</a>
P05	Bryan	<a href="#">View</a>
P06	Aaron	<a href="#">View</a>
P07	Jake	<a href="#">View</a>
P08	Elaine	<a href="#">View</a>
P09	Lily	<a href="#">View</a>

Figure 41 Treatment Record Page Wireframe

This is the treatment record page; users can direct to here by clicking the option on the side navigation bar. In this page, it will show a list of patient information and beside each patient there will be a view button for viewing the record of the selected patient.



- Schedule
- Appointment
- Treatment Record
- Accounts
- Logout

### Treatment Record- Ali

Appointment ID	Date	Details	
A03	12/1/2022	<a href="#">View</a>	<a href="#">Edit</a>
A10	19/1/2022	<a href="#">View</a>	<a href="#">Edit</a>
A23	25/1/2022	<a href="#">View</a>	<a href="#">Edit</a>
A42	12/2/2022	<a href="#">View</a>	<a href="#">Edit</a>

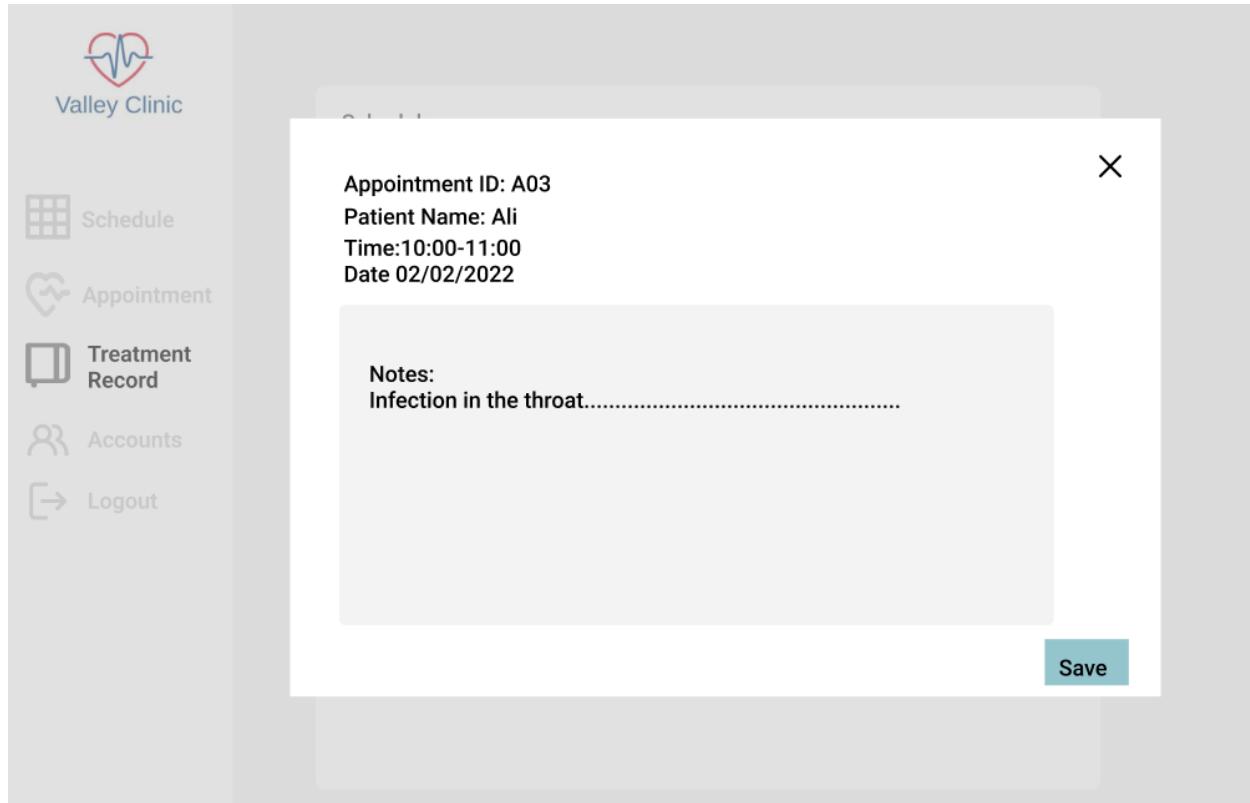
Figure 42 Treatment Record Page 2 Wireframe

This is the page after the user clicking on the view button beside the patient details, it will come to this page showing all past appointment record of the selected patient. Beside the information of the appointment there are two button which is view and edit. The view button enable the users view the details of the treatment record of the selected appointment, however the edit button enable the users to edit information of it.

The screenshot shows a mobile application interface for 'Valley Clinic'. On the left is a vertical navigation bar with icons and labels: 'Schedule' (grid icon), 'Appointment' (heart icon), 'Treatment Record' (document icon), 'Accounts' (person icon), and 'Logout' (exit icon). The main area displays a modal window with a white background and a close button ('X') in the top right corner. Inside the modal is a table with the following data:

Appointment ID:	A03
Patient Name:	Ali
Time:	10:00-11:00
Date:	02/02/2022
Notes:	Infection in the throat..... .....

This is the modal that will appear after the user click on the view button, a modal with a table showing all the treatment record details,



This is the modal that will appear after the users click on the edit button, an input box that allow users to edit or change the details. There is also a save button down there to save the changes the user had made.

### 15.3.4 Doctor Account Page

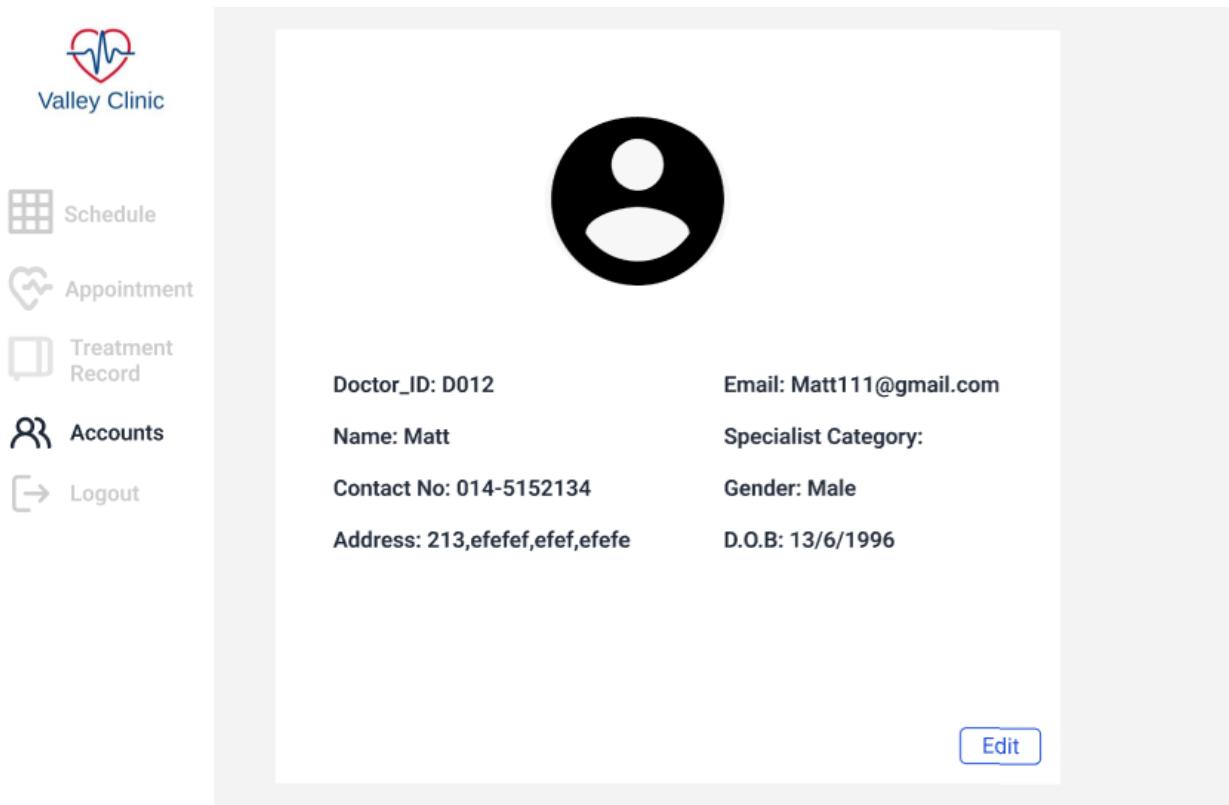


Figure 43 Doctor Account Page Wireframe

This is the doctor account page; user can direct to here by clicking the account button on the side navigation bar. This page will show all the details information of the user. User can check and edit their information by clicking the edit button down there.



-  Schedule
-  Appointment
-  Treatment Record
-  Accounts
-  Logout

Profile Picture :

Doctor\_ID:

Name:

Contact No:

Address:

Email:

Specialist Category:

Gender:

D.O.B:

Figure 44 Doctor Edit Account Profile Page Wireframe

This is the page where user will be directed after clicking the edit button in the account page. Multiple input box will be provided to users to fill in. After finish filling up the form, users can click on the save button down there to edit and save their information into database.

### 15.3.5 Doctor Logout Page

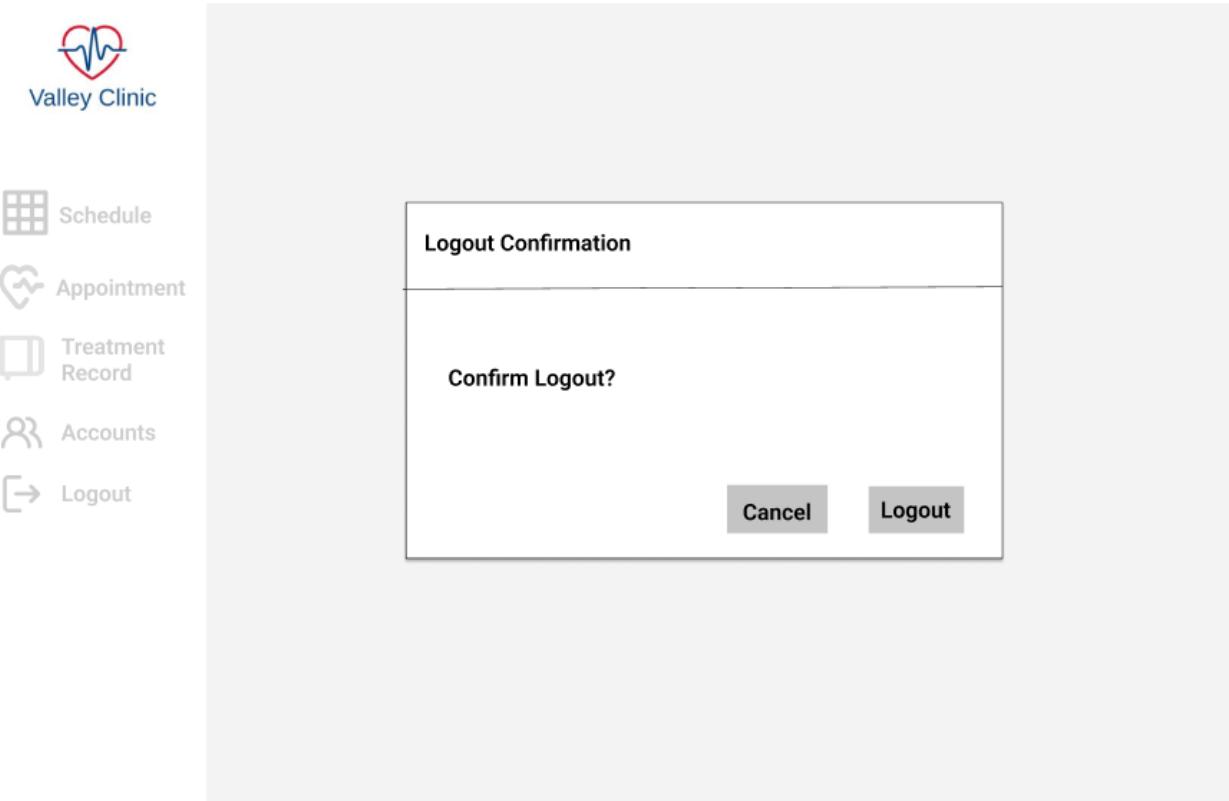
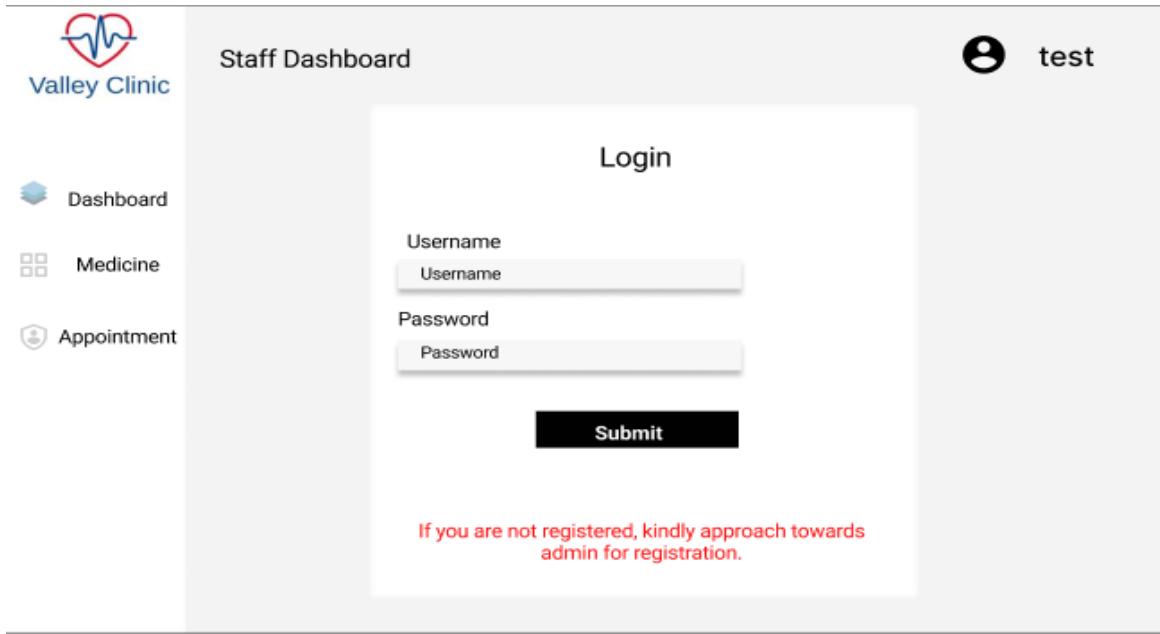


Figure 45 Doctor Logout Page Wireframe

This is the doctor logout page, after users clicking on the logout button on the side navigation bar a logout confirmation model will appear asking for user logout confirmation.

## 15.4 Staff Module

### 15.4.1 Staff Login

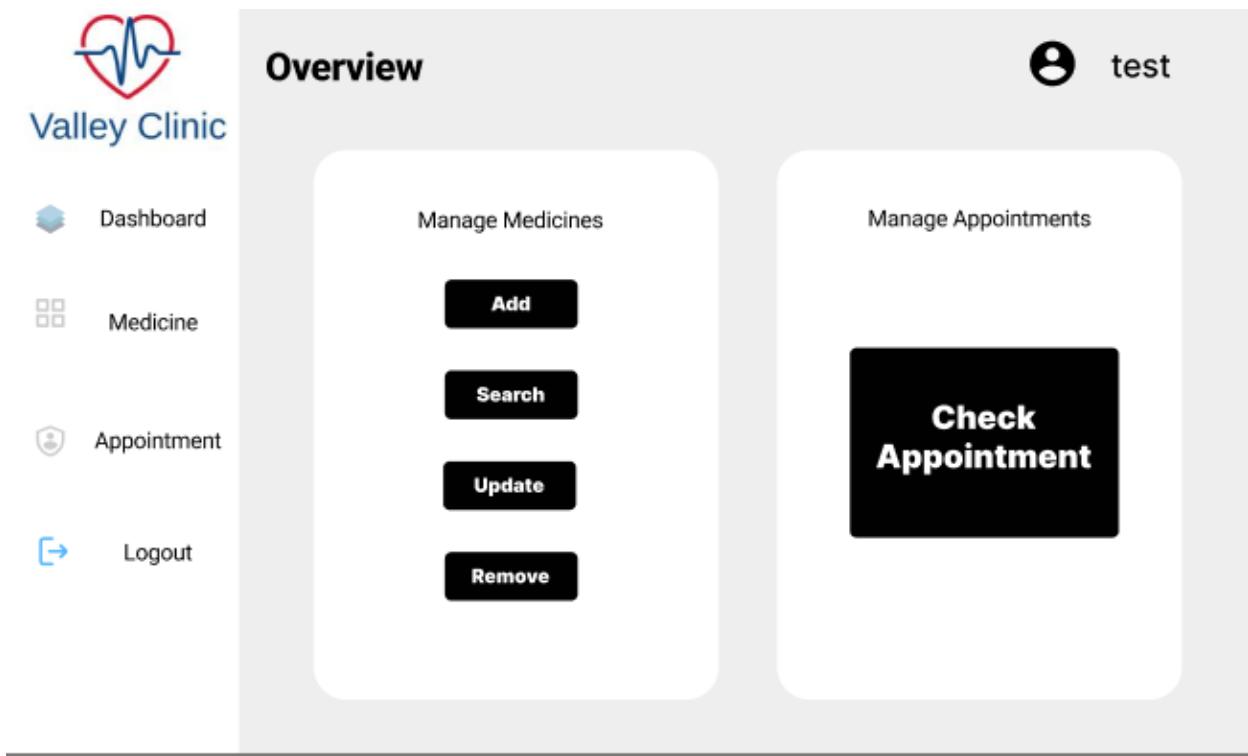


The screenshot shows the 'Staff Dashboard' interface for Valley Clinic. On the left, there's a sidebar with icons for 'Dashboard', 'Medicine', and 'Appointment'. At the top right, a user profile is shown with the name 'test'. The main area is titled 'Login' and contains fields for 'Username' and 'Password', both with placeholder text ('Username' and 'Password'). Below these is a 'Submit' button. A red message at the bottom states: 'If you are not registered, kindly approach towards admin for registration.'

*Figure 46: Staff Login Page*

The figure above shows the login page from the staff side when logging into Valley Clinic system.

#### 15.4.2 Staff Dashboard Overview



*Figure 47 : Valley Clinic Staff Dashboard*

Figure 47 shows the overview dashboard page for the staff and what functions are available to the staff side of the system. Moreover, the staffs can click on the buttons directly which will then bring them to the pages directly.

### 15.4.3 Add Medicine

The screenshot shows a web-based application interface for 'Valley Clinic'. In the top left corner is a red heart icon with a blue ECG line. To its right, the text 'Valley Clinic' is displayed. In the top right corner, there is a user icon with the name 'test' next to it. The main title 'Add Medicines' is centered at the top of the page. On the left side, there is a vertical navigation menu with the following items: 'Dashboard' (represented by a dashboard icon), 'Medicine' (represented by a grid icon), 'Appointment' (represented by a calendar icon), and 'Logout' (represented by a person icon). The central area contains five input fields labeled 'Medicine ID', 'Description', 'Medicine Name', 'Quantity To Buy', and 'Price', each with a corresponding text input box. At the bottom right of this central area are two blue buttons: 'Add Item' and 'Cancel'.

*Figure 48 : Add Medicine Page*

On the add medicine page, the staff can add medicine into the system by entering the required information to store into the system's database.

#### 15.4.4 Search Medicine

The image shows a mobile application interface for 'Valley Clinic'. On the left is a sidebar with a heart icon and the text 'Valley Clinic'. The main area has a title 'Search Medicines' and a user icon labeled 'test'. It contains three search fields: 'Enter Medicine ID :', 'Medicine Name :', and 'Quantity In Stock :'. Below the fields are 'Back' and 'Cancel' buttons. The sidebar also includes links for 'Dashboard', 'Medicine', 'Appointment', and 'Logout'.

Search Medicines

test

Enter Medicine ID :

Medicine Name :

Quantity In Stock :

Back Cancel

Dashboard Medicine Appointment Logout

Valley Clinic

Figure 49 : Search Medicine

In this page, the staffs can search all available medicine in the stock inventory by entering the medicine ID.

#### 15.4.5 Edit Medicine

The screenshot shows a web-based application interface for managing medicine inventories. On the left, there is a vertical sidebar with the 'Valley Clinic' logo at the top. Below the logo are four menu items: 'Dashboard' (represented by a blue icon), 'Medicine' (represented by a blue icon), 'Appointment' (represented by a blue icon), and 'Logout' (represented by a blue icon). At the top right of the main content area, there is a user profile icon labeled 'test'. The main title 'Update Medicines' is centered above a form. The form contains four input fields: 'Enter Medicine ID :' with an input field, 'Search' button, 'Medicine Name :' with an input field, 'Add / Subtract Quantity :' with an input field, and 'Balance in Quantity :' with an input field. At the bottom right of the form are two buttons: 'Update' and 'Cancel'.

*Figure 50 : Edit Medicine*

On this page, the staff can edit the medicines table by entering the medicine ID and clicking on whether to increase or decrease the number of quantities they would likely have.

#### 15.4.6 Delete Medicine

The screenshot shows the Valley Clinic software interface. On the left is a sidebar with icons for Dashboard, Medicine, Appointment, and Logout. The main area has a title 'Remove Medicines' and a user 'test'. It contains three input fields: 'Enter Medicine ID :', 'Medicine Name :', and 'Number of Quantity :'. Below these is a 'Search' button. At the bottom right are 'Remove' and 'Cancel' buttons.

Valley Clinic

Remove Medicines

test

Dashboard

Medicine

Appointment

Logout

Enter Medicine ID :

Medicine Name :

Number of Quantity :

Search

Remove

Cancel

*Figure 51 : Delete Medicine*

The staffs can delete a certain medicine if there is no use of the said medicine in the clinic by entering the medicine ID and the quantity that they would like to delete from the system's database.

#### 15.4.7 Check Appointment



Valley Clinic

Dashboard Medicine Appointment Logout

test

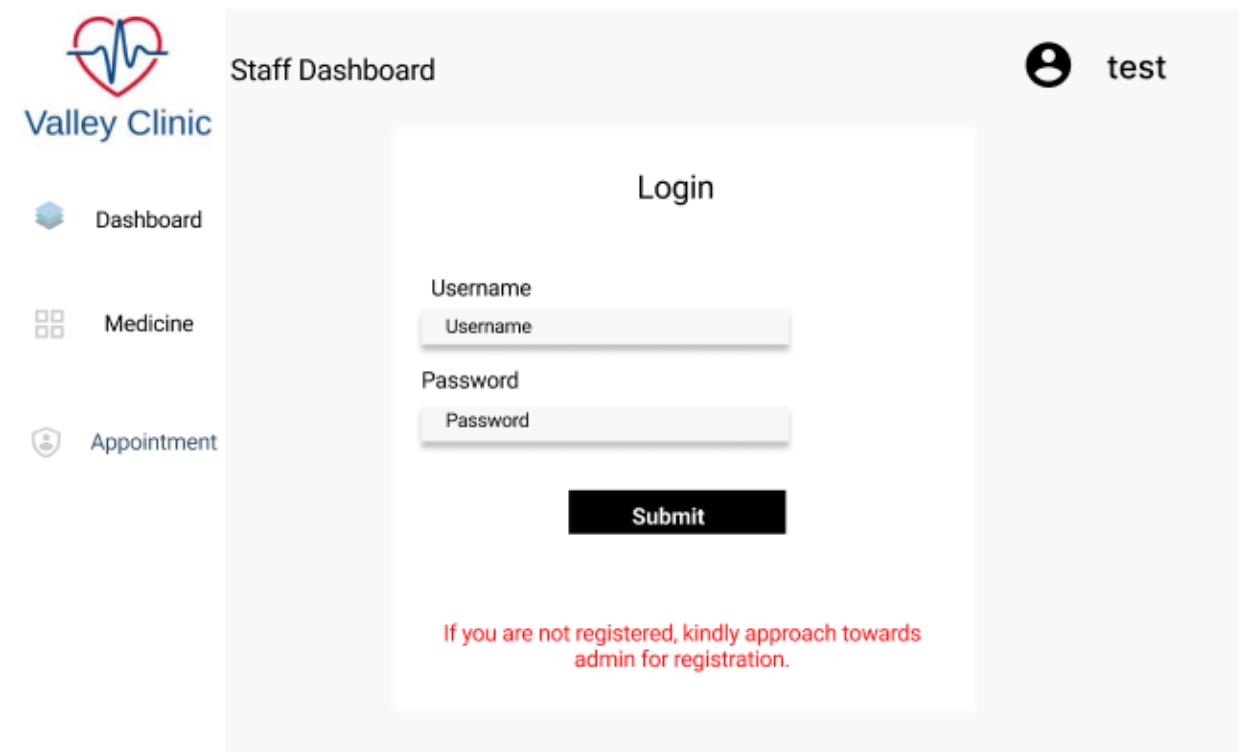
AppointmentID:	<input type="text"/>	Search
AccountID:	<input type="text"/>	
DoctorID:	<input type="text"/>	
Appointment Date:	<input type="text"/>	
Appointment Slot:	<input type="text"/>	
Appointment Status:	<input type="text"/>	

Back Cancel

*Figure 52 : Check Appointment*

In this page, the staff can check on the appointments that are listed for that day and is able to keep track of all the appointments that has been booked for each doctor in the clinic.

#### 15.4.8 Staff Logout



*Figure 53: Successfully Signed Out*

The figure above shows the page when a staff has successfully signed out from their Valley Clinic Staff Login Page which will then be redirected to the login page of the staff system.

## 16.0 Report Designs and User Manual

### 16.1 Customer Module

#### 16.1.1 Navigation Bar

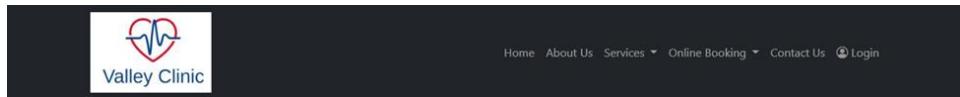


Figure 12 Navigation Bar

One of the main component which can be seen consistent across multiple webpages which mainly targeted for the use of our customers would be the navigational structure or the navigation bar. Starting off from the left side of the navigation bar locates the logo of our clinic which would be Valley Clinic. Additionally, the main focus of the navigation bar would be the tabs or link provided to the customer which allow for quick navigation and easy accessibility where customers are given multiple tabs such as the homepage, about us, services, online booking and contact us page. Alongside that the users can also opt to login into the system to access certain webpages which are lock for registered users only, for example, the online booking page which allows the user to make booking based on their preferred preferences such as time, date, and doctor of choice. Additionally, multiple drop downs are also provided for further classification of that particular section such as the services tab where the user can opt to view the treatments provided by the clinic and another option where the user can opt to view our medical teams which is located under the “our doctor” tab.

#### 16.1.2 Footer

About Us	Contact Us	Follow Us
We at Valley Clinic provides medical services to all patients with a sense of security while welcoming you with a smile on our face. No matter the seriousness of your condition, we at Valley Clinic are well equip with latest medical equipment along with experienced doctors. Your health is our priority, pay us a visit at Valley Clinic for your medical needs.	For more information:  2, Jalan Alam Jaya 15, Taman Alam Jaya, 43200 Cheras, Selangor +60 3-9226 9999 <a href="mailto:help@valleycl.com.my">help@valleycl.com.my</a>	<a href="#">Facebook</a> ValleyClinic <a href="#">Twitter</a> ValleyClinicMY <a href="#">Instagram</a> @ValleyClinicMY

Figure 13-Footer

Alongside the navigation bar, one of the most significant component of our webpage includes the footer section where additional information such as the Clinic info, contacts, and social links are all located in the footer. Additionally, these information are also available in their respective tabs which the user can navigate to by using the navigational structure located at the top of each

webpage. The need for consistent navigation bar and footer provides clarification and puts the user at ease knowing that they are visiting the clinic's webpage and when they navigate throughout the webpages provided, they won't be met with discomfort, worried that they might not still be on the same website.

### 16.1.3 Customer Homepage

**VALLEY CLINIC MEDICAL INSTITUTION**

PAY US A VISIT!

Why choose us?

Our patient's safety and well-being is our number 1 priority. Along with our top quality medication product, we provide top tier treatment from our certified doctors with years of experience. Pay us a visit and we will ensure that you leave our doorstep as good as new.

**Service Highlights**

- Experienced doctors and nurses
- Hygienic and comfortable environment
- Latest medical equipment
- Booking and appointment services

**Medical Consultation Process**

- Make an Appointment
- Physical Confrontation
- Contact Us!

**About Us**

We at Valley Clinic provides medical services to all patients with a sense of security while welcoming you with a smile on our face. No matter the seriousness of your condition, we at Valley Clinic are well equip with latest medical equipment along with experienced doctors. Your health is our priority, pay us a visit at Valley Clinic for your medical needs.

**Contact Us**

For more information:

- 2, Jalan Alam Jaya 15, Taman Alam Jaya, 43200 Cheras, Selangor
- +60 3-9226 9999
- [help@valleyct.com.my](mailto:help@valleyct.com.my)

**Follow Us**

- [Facebook](#) ValleyClinic
- [Twitter](#) @ValleyClinicMY
- [Instagram](#) @ValleyClinicMY

Figure 14 Customer Homepage

Upon customer visit, the user will be prompt with the homepage where throughout the whole website which are targeted for customer's viewing pleasure are supplied with a navigational bar and a footer section which both are consistently matched in all customer related webpages. Additionally, if the user opt for viewing the homepage tab which by default would be automatically be the first webpage where the user are prompt to, the user could select the "Home" tab located on the navigational bar. Furthermore, if the user wishes to seek further information on the current webpage they are in, namely the Homepage, the user can view the entire homepage by scrolling though the webpage which display a collection of information. For example, in the main content of the homepage, the user are provided with a carousel which display informative images for viewing pleasure while providing user interaction. Alongside with that, a brief introduction to the company are also given in the form of text, services highlight, and medical consultation process are all available information accessible in the homepage.

#### 16.1.4 About Us

The screenshot shows the Valley Clinic website. At the top, there is a dark header bar with the Valley Clinic logo (a red heart with a blue ECG line) and the text "Valley Clinic". To the right of the logo are navigation links: Home, About Us, Services (with a dropdown arrow), Online Booking (with a dropdown arrow), Contact Us, and a login link. Below the header is a large, blurred background image of a medical office. Overlaid on this image is a quote in white text: "'The greatest wealth is health'" followed by "Virgil". At the bottom of the screenshot, the heading "About Us" is visible, along with some placeholder text in a smaller font.

Lore ipsum dolor, sit amet consectetur adipisicing elit. Optio, vitae cupiditate aspernatur sit, quas sed architecto ipsum possimus inventore illum provident harum deserunt qui facere molestiae eaque facilis eum quasi.



## Our Mission

To provide medical services for the community starting from rural locations or less advanced community without discrimination over individual social status. All of which are in goals of providing an affordable healthcare service to the general public.

## Our Vision

To be the leading corporate and leading healthcare provider in Malaysia, all while providing health care services that are ethical and affordable to the general public.



### About Us

We at Valley Clinic provides medical services to all patients with a sense of security while welcoming you with a smile on our face. No matter the seriousness of your condition, we at Valley Clinic are well equip with latest medical equipment along with experienced doctors. Your health is our priority, pay us a visit at Valley Clinic for your medical needs.

### Contact Us

For more information;

2, Jalan Alam Jaya 15, Taman Alam Jaya,  
43200 Cheras, Selangor  
+60 3-9226 9999  
[help@valleycl.com.my](mailto:help@valleycl.com.my)

### Follow Us

[Facebook](#) ValleyClinic  
[Twitter](#) @ValleyClinicMY  
[Instagram](#) ValleyClinicMY

Figure 14 About us page

One of the main section which are accessible to the user consist of the about us page, where the user can opt to view more information on the clinic's background. For example, the clinic provides information on the current state of their clinic and the medical process and treatment given to their customer through the about us page. More information on the clinic business process are also given such as information of the clinic's mission and their vision.

## 16.1.5 Contact Us



Valley Clinic

Home About Us Services ▾ Online Booking ▾ Contact Us Login



### Contact Us



Klinik Mediviron  
2, Jalan Alam Jaya 15, Taman Alam Jaya, 43200 Cheras, Selangor  
3.3 ★★★★☆ 16 reviews  
[View larger map](#)

Saujana Golf & Country Club, Pacific Place 24JAM...  
Subang Parade, Sunway Lagoon, SEKSYEN 51A, TAMAN OUG, Subang Jaya, Puchong Avenue, Puchong, Seri Kembangan, Ampang Jaya, Pandan Indah, Cheras, Hulu Langat, Sungai Tekala, Sungai Long, Bandar Baru Sungai Long, Bandar Mahkota Cheras, Semenyih Ecoventure Resort & Recreation, Sangkot Tree Kg Paya Seberang Desa Tua, Kolam Air Panas Batu 16, Air Terjun Siong.

Get in Touch With Us!

#### Our Socials

[Facebook](#) [Instagram](#) [Twitter](#)

#### Contacts

+60 3-9226 9999  
[help@valleycl.com.my](mailto:help@valleycl.com.my)

#### Location

2, Jalan Alam Jaya 15, Taman Alam Jaya, 43200 Cheras, Selangor

## Contact Us

---

Name

Contact Number

Email

Comment

---

[Reset](#) [Submit](#)

**About Us**

We at Valley Clinic provides medical services to all patients with a sense of security while welcoming you with a smile on our face. No matter the seriousness of your condition, we at Valley Clinic are well equip with latest medical equipment along with experienced doctors. Your health is our priority, pay us a visit at Valley Clinic for your medical needs.

**Contact Us**

For more information;

2, Jalan Alam Jaya 15, Taman Alam Jaya,  
43200 Cheras, Selangor

+60 3-9226 9999  
[help@valleycl.com.my](mailto:help@valleycl.com.my)

**Follow Us**

[Facebook](#) [Twitter](#) [Instagram](#)

*Figure 15 Contact us page*

In the contact us page, which is also accessible through the navigation bar located above all customer targeted webpages. The contact us page provide user with information on the physical location of which the clinic is located. This is shown through the google map provided in the contact us page where the user can interact with to view the location of the physical location of the clinic. Additionally, the users are also provided with information on the contact details such as the social links, contact number, email address, as well as the address of the clinic where the user can contact the clinic if they wish for any of their enquiries to be answered. Lastly, the user can also send a short message to the clinic staff if they posses any inquiries they wish to be answered. For example, the user are to enter their name, contact number, email address as well as a short message which the user wish to message the clinic administrator. Once all the information are provided are given and met with the form validation criteria, the user can opt to submit the form which conveys the information to the staff administrators or the user can reset the form which removes the information provided where the user can reenter their information as well as message all over again if the provided information are not met with users expectations.

## 16.1.6 Treatment and Services

The screenshot shows the 'Treatment & Services' page of the Valley Clinic website. At the top, there is a navigation bar with links for Home, About Us, Services, Online Booking, Contact Us, and Login. Below the navigation is a large image of a modern operating room. The main heading 'Treatment & Services' is centered above a search bar with a placeholder 'Keyword' and a magnifying glass icon. Below the search bar is a grid of three service categories: 'Asthma', 'Emergency Medicine', and 'Anesthesiology', each with a small thumbnail image. A navigation indicator '1 >>' is positioned below the grid. Below the grid are six smaller service icons: Birthing Care (two people), Healthcare (heart), Emergencies (ambulance), Family Medicine (person with stethoscope), Cancer Care (cancer ribbon), and Orthopedics (bone). A section titled 'Services & Equipments' features a photo of a doctor examining a young girl with a stethoscope. To the right of the photo is a text block about hygiene and service. Another section titled 'Our Equipment' shows four circular images of medical equipment: an operating room, a monitor, a心率机, and a ultrasound machine.

### About Us

We at Valley Clinic provides medical services to all patients with a sense of security while welcoming you with a smile on our face. No matter the seriousness of your condition, we at Valley Clinic are well equip with latest medical equipment along with experienced doctors. Your health is our priority, pay us a visit at Valley Clinic for your medical needs.

### Contact Us

For more information:

2, Jalan Alam Jaya 15, Taman Alam Jaya,  
43200 Cheras, Selangor

+60 3-9226 9999

help@valleycl.com.my

### Follow Us

ValleyClinic

@ValleyClinicMY

@ValleyClinicMY

Figure 16 Treatment and Services page

Under treatment and services webpage, the user are met with a display of information on the available treatment and services provided by Valley Clinic. To start off user's viewing process,

the user can opt to search for specific treatment or services provided by the clinic through the search bar where the user will interact with by entering the desired search keyword such as “asthma” if the result matches with our database where all information on the treatment are similar to the one mentioned by the user, the result will be displayed right below the search bar. Additionally, if more than one result are to be displayed, a pagination which allows the user to navigate between the results without clogging the display on the user screen are also provided. Furthermore, more information on the services and equipment used during the treatment process are also provided with the help of visual aids such as the one being displayed under “Services and Equipment” which shows a gallery of images regarding all the equipment available in the clinic. Aside from that, the user can also view information regarding the available treatment by clicking on the link or rather the name of the particular treatment of choice under the result being displayed by the search feature.

### 16.1.7 Medical Services detail

The screenshot shows a medical services detail page for the condition 'Asthma'. At the top, there's a navigation bar with the Valley Clinic logo, a heart icon, and links for Home, About Us, Services, Online Booking, Contact Us, and Login. Below the navigation is a photograph of a medical operating room with two large surgical lights. The word 'Asthma' is overlaid on the image. The main content area is divided into several sections:

- Description:** A box containing the text: "No additional information to be displayed. Please refer to your nearest healthcare center for further diagnosis."
- Consultation Fees:** A box showing the fee as "RM 100.00~ Per Consultation".
- Specialist Doctors:** A section featuring two doctor profiles: "Errick Mattinson" and "Randolf Selbach", each with a circular icon of a doctor wearing a stethoscope.
- About Us:** A box describing Valley Clinic's mission to provide medical services with security and a smile, equipped with latest medical equipment and experienced doctors.
- Contact Us:** Information including address (2, Jalan Alam Jaya 15, Taman Alam Jaya, 43200 Cheras, Selangor), phone number (+60 3-9226 9999), and email (help@valleycl.com.my).
- Follow Us:** Social media links for Facebook (@ValleyClinic), Instagram (@ValleyClinicMY), and Twitter (@ValleyClinicMY).

Figure 17 Medical Services detail Page

After the user had chosen the medical services, they wish to view more in detailed, the user will be navigated to the medical services detail page where information regarding the medical condition are displayed. For example, some of the information being displayed include a brief description on the medical condition, the consultation fee which will be charged if the user wish to undergo treatment for the particular disease, as well as the specialized doctors.

## 16.1.8 Our Doctors

Home About Us Services ▾ Online Booking ▾ Contact Us Login

### Our Doctors

To replace any placeholder text (such as this), just click it and start typing. We think this paragraph makes a great statement just as it is. But if you'd like to try a bit of customizing to make it your own, you can change the fonts with just a click.

Lawry Barracks  
Allergists

Nolana Dowyer  
Anesthesiologists

Lorrayne Olle  
Cardiologists

Erick Mattinson  
Physician

Randolph Selbach  
Physician

#### About Us

We at Valley Clinic provides medical services to all patients with a sense of security while welcoming you with a smile on our face. No matter the seriousness of your condition, we at Valley Clinic are well equip with latest medical equipment along with experienced doctors. Your health is our priority, pay us a visit at Valley Clinic for your medical needs.

#### Contact Us

For more information;

2, Jalan Alam Jaya 15, Taman Alam Jaya,  
43200 Cheras, Selangor

+60 3-9226 9999

help@valleycl.com.my

#### Follow Us

[Facebook](#) [Twitter](#) [Instagram](#)

Figure 18 Our Doctor Page

In the doctor page which the user can access through the navigation bar, information regarding the available doctors and their respective specialist are displayed in the doctor page.

### **16.1.9 User Notification on requirement (Login/Sign Up)**



*Figure 19 User Alert on required login*

Aside from the previous mentioned webpages such as the doctor page, treatment and services, homepage, about us page and contact us page, the user will be required to posses a registered account in order to access some features which are reserved for registered users only. For instance, the booking page and patient's booking history records which requires the user to posses a registered account.

### **16.1.10 User Sign In**

A screenshot of a user sign-in interface. At the top center is a "Sign In" button. Below it is a "Username:" label with an input field containing "Username". Below that is a "Password:" label with an input field containing "Password". At the bottom is a large purple "Sign In" button. Below the button is the text "Not a member? [Click here to Sign Up!](#)".

*Figure 20 User Sign in Interface*

**Sign In**

Incorrect Username or Password!

Username:

Password:

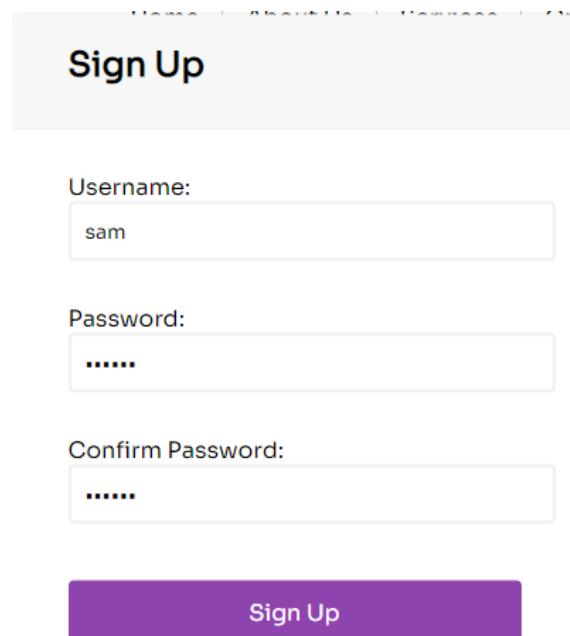
**Sign In**

Not a member? [Click here to Sign Up!](#)

*Figure 21 Invalid Username / Password*

In the user sign in interface, the user will be given the option to sign in via username and password or if user wish to create a new account, they can do so by clicking on the hyperlink given below. If the user entered an incorrect combination of username or password. A prompt will be shown, notifying the user on the error which occurred during the sign-in process.

### 16.1.11 User Sign-Up



A screenshot of a user sign-up form titled "Sign Up". It includes fields for "Username" (containing "sam"), "Password" (containing "\*\*\*\*\*"), and "Confirm Password" (containing "\*\*\*\*\*"). A purple "Sign Up" button is at the bottom.

Username:  
sam

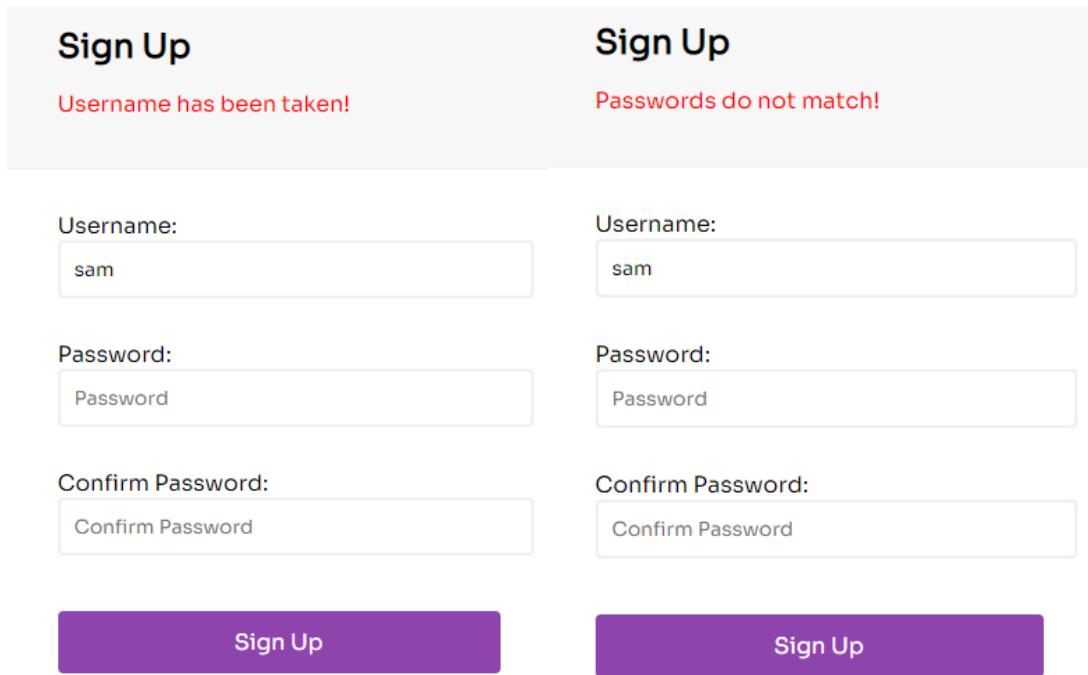
Password:  
\*\*\*\*\*

Confirm Password:  
\*\*\*\*\*

Sign Up

Already a member? [Click here to Sign In!](#)

Figure 22 User Sign Up



The image shows two identical "Sign Up" forms side-by-side, each displaying an error message below the respective field.

**Left Form Error:** "Username has been taken!"

**Right Form Error:** "Passwords do not match!"

Both forms have the same layout:

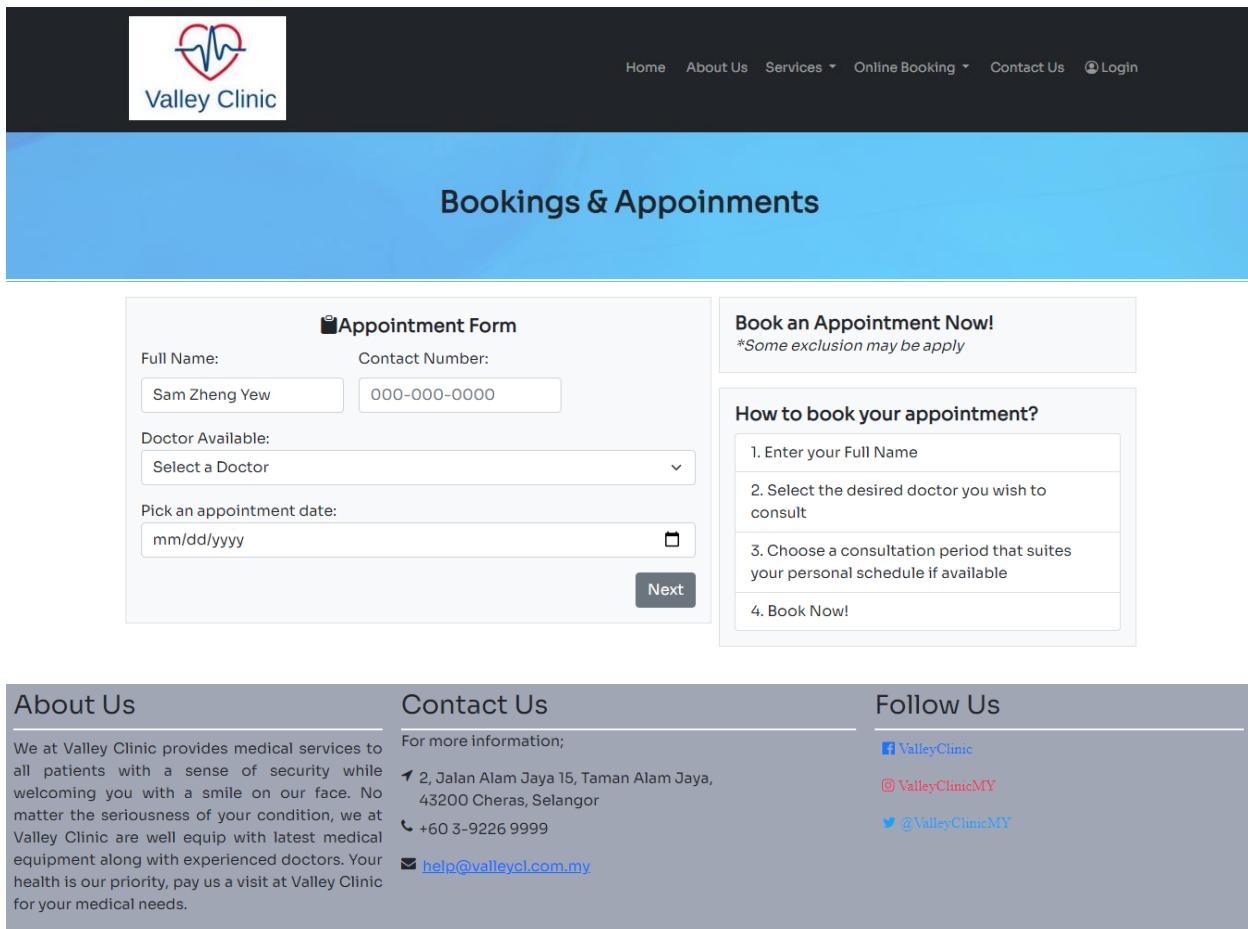
- Username:** Input field containing "sam".
- Password:** Input field containing "Password".
- Confirm Password:** Input field containing "Confirm Password".
- Sign Up** button at the bottom of each form.

Already a member? [Click here to Sign In!](#)

Figure 23 Error Messages during Sign Up

If the user opt to sign-up, they will be prompt with the sign-up interface as shown in Figure 23 which consist of a form input. In the given form, the user are to provide their desired username, password as well as a validation on the confirm password in order to successfully create an account. Aside from that, the registration form also includes form validation if the user entered a combination of username and password which previous or other users had already taken, an error will be prompt to notify the user of such situation. The user will have to reenter a new username and password in this case.

### 16.1.11 Booking and Appointment



The screenshot shows the 'Bookings & Appointments' page for Valley Clinic. At the top, there's a navigation bar with the clinic's logo (a heart with a pulse line) and the text 'Valley Clinic'. The navigation links include Home, About Us, Services, Online Booking, Contact Us, and a Login button. Below the navigation is a blue header section with the text 'Bookings & Appointments'. The main content area is divided into two sections: 'Appointment Form' on the left and 'Book an Appointment Now!' on the right. The 'Appointment Form' section contains fields for Full Name (Sam Zheng Yew), Contact Number (000-000-0000), Doctor Available (Select a Doctor dropdown), and Pick an appointment date (mm/dd/yyyy). A 'Next' button is located at the bottom right of this section. The 'Book an Appointment Now!' section has a heading and a note about exclusions. It also includes a 'How to book your appointment?' section with four numbered steps: 1. Enter your Full Name, 2. Select the desired doctor you wish to consult, 3. Choose a consultation period that suites your personal schedule if available, and 4. Book Now! At the bottom of the page, there are three columns: 'About Us', 'Contact Us', and 'Follow Us'. The 'About Us' column contains a paragraph about the clinic's services. The 'Contact Us' column provides address, phone number, and email. The 'Follow Us' column lists social media links for Facebook, Instagram, and Twitter.

Figure 24 Booking and Appointment Page

One of the functions which are only accessible to registered users only consist of the booking and appointment page where the user are given the option to make an appointment based on their preference date and time. Starting off with the booking page, the user will be required to enter their personal information such as their full name and contact info. Additionally, they are

also required to choose their desired doctor they wish to visit by selecting one of the given option in the dropdown menu. Furthermore, the user can also choose their date of appointment as well as the available time slots. Additionally, information regarding the booking process are also listed on the sidebar, located right next to the appointment form. By providing the appointment date, the system will check for available slots on that particular day and if there are slots which are not available, they will be disabled for selection. Once the users are satisfied with their provided information, the user can navigate to the next page of the booking process by clicking on the submit button.

**Appointment Form**

Full Name: Sam Zheng Yew Contact Number: 0129239825

Doctor Available: Lawry Barracks

Pick an appointment date: 03/13/2022

[Next](#)

**Book an Appointment Now!**  
\*Some exclusion may be apply

**How to book your appointment?**

1. Enter your Full Name
2. Select the desired doctor you wish to consult
3. Choose a consultation period that suites your personal schedule if available
4. Book Now!

Available Time Slots:

08.30	09.00	09.30	10.00	10.30	11.00	11.30	12.00	
12.30	13.00	13.30	14.00	14.30	15.00	15.30	16.00	16.30
17.00	17.30	18.00	18.30	19.00				

[Submit](#)

Figure 25 Example of Booking process

### 16.1.12 Confirm Appointment Details

The screenshot shows a web-based appointment booking form titled 'Confirm Appointment Details'. It includes fields for 'Full Name' (Sam Zheng Yew) and 'Contact Number' (0129239825). A dropdown menu for 'Doctor Available' shows 'Lawry Barracks'. The date '03/13/2022' is selected for 'Pick an appointment date'. Below these, a grid of time slots is displayed, with '08.30' highlighted in green. Other slots include 09.00, 09.30, 10.00, 10.30, 11.00, 11.30, 12.00, 12.30, 13.00, 13.30, 14.00, 14.30, 15.00, 15.30, 16.00, 16.30, 17.00, 17.30, and 18.00. A blue 'Confirm Booking' button is at the bottom.

Figure 26 Confirm Booking Details

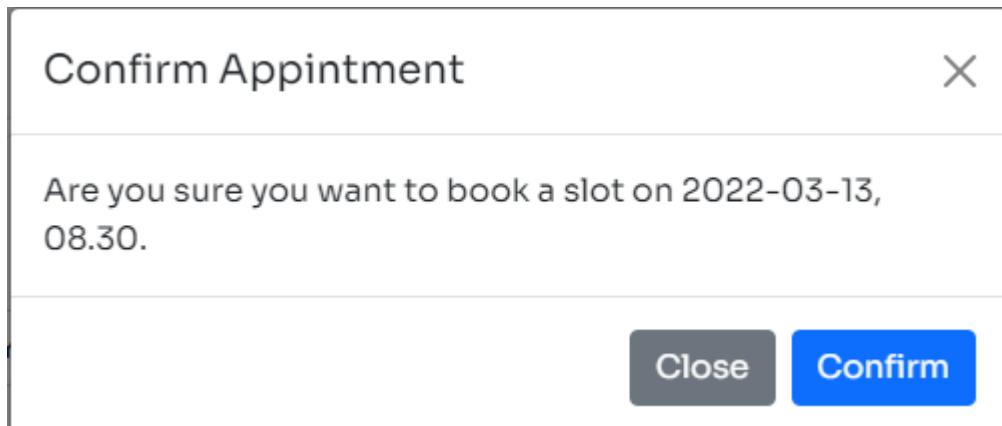


Figure 27 Notification on user confirmation on the provided information

After user had submitted their appointment details, the user will be navigated to an additional page, where information on the provided appointment details are displayed and if the user wants to change any of the information that was previously given, they can do so by changing them before finally submitting the form for entry. A pop-up notification will ask the user for confirmation whether they wish to submit their booking. If the user chose the close option, they will be redirected back to the confirmation page whereas if the user opt to submit their booking information, they can do so by clicking on the confirm button.

### 16.1.13 Payment Gateway

The screenshot shows the Valley Clinic website's payment gateway page. At the top, there is a black header bar with the Valley Clinic logo (a red heart with a white ECG line) and navigation links: Home, About Us, Services, Online Booking, Contact Us, and Login.

The main content area has a yellow-to-pink gradient background. It features a title "Payment Gateway" and a section titled "Booking Details / Invoice".

In the "Booking Details / Invoice" section, it displays:

- Sam Zheng Yew**
- Date Issued: 03-12-2022**
- Total Charged: RM 50**
- Appointment Details:**
  - Appointment Doctor: Lawry Barracks
  - Appointment Date: 2022-03-13
  - Appointed Time: 08.30

Below this, there are small terms and conditions and contact information, followed by two buttons: "Pay Later" and "Pay now!".

To the right, there are two boxes:

- Make Payment Now!**
  - Check your information
  - Select your desire payment method
  - Fill in your information as labelled
  - Don't want to pay now? No problem, walk in and pay us once you're done with your treatment
- When making a payment/booking...**
  - Please take a screenshot of the following Invoice as evidence.
  - You may make prior payment for consultation cost, if not you may pay later as well.
  - Keep in mind, consultation

Figure 28 Payment Details

After which the user had confirm their booking details and all final changes had been made, the user will be navigated to the payment page where information that was previously provided such as the customer's name, doctor of choice, appointment date, and appointment slots will be displayed. Additionally, users are given the option to choose between paying later or paying now via Credit Card or cash. If the user choose to pay later, their booking process will be complete (Figure 33) whereas if the user opt to pay now, the system will ask the user whether they wish to pay via online banking or by cash (Figure 30). If the user choose to pay via online banking, they are required to provide additional information regarding their personal information such as the credit card number, credit card name, the expiry date and the CVV.

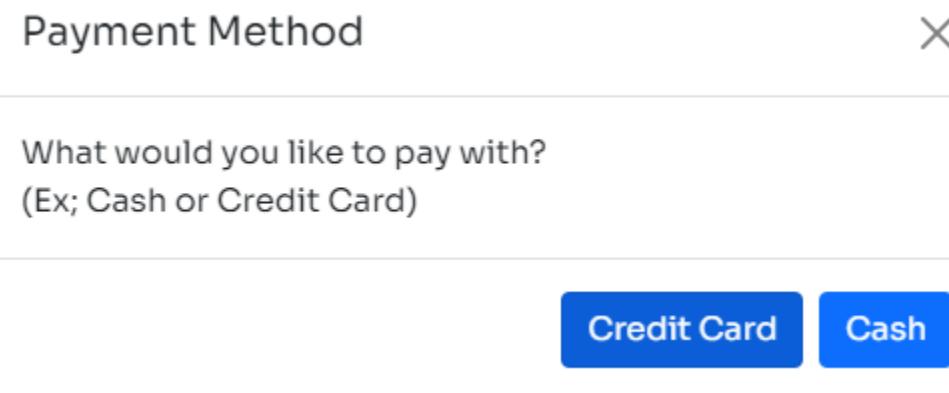


Figure 29 Payment Method of choice

Cash
Credit Card

Cash Payment

Total Amount Due: 50.00

An invoice will be sent via email.  
Please proceed to receptionist counter on the day of appointment to validate appointment booking with payment.

[Proceed to Checkout](#)

#### About Us

We at Valley Clinic provides medical services to all patients with a sense of security while welcoming you with a smile on our face. No matter the seriousness of your condition, we at Valley Clinic are well equip with latest medical equipment along with experienced doctors. Your health is our priority, pay us a visit at Valley Clinic for your medical needs.

#### Contact Us

For more information:

- 📍 2, Jalan Alam Jaya 15, Taman Alam Jaya,  
43200 Cheras, Selangor
- 📞 +60 3-9226 9999
- ✉️ [help@valleycl.com.my](mailto:help@valleycl.com.my)

#### Follow Us

Facebook ValleyClinic  
Twitter @ValleyClinicMY  
Email @ValleyClinicMY

Figure 30 Payment by cash

Cash
Credit Card

Credit Card

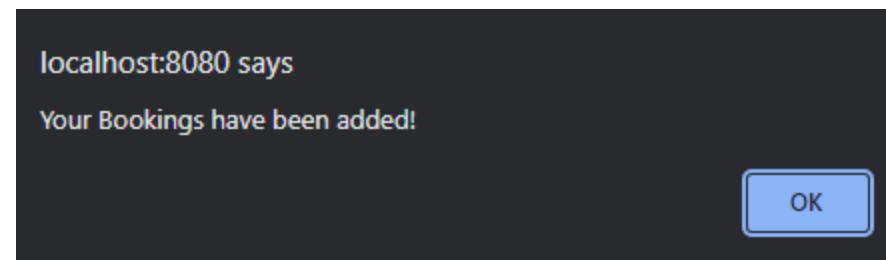
Card Number:

Card Name:

Expiry Date:  /  Security Code:

[Proceed to Checkout](#)

Figure 31 Payment Via Credit Card



*Figure 32 Notification on successfully book an appointment*

## 16.2 Administrator Module

### 16.2.1 Login

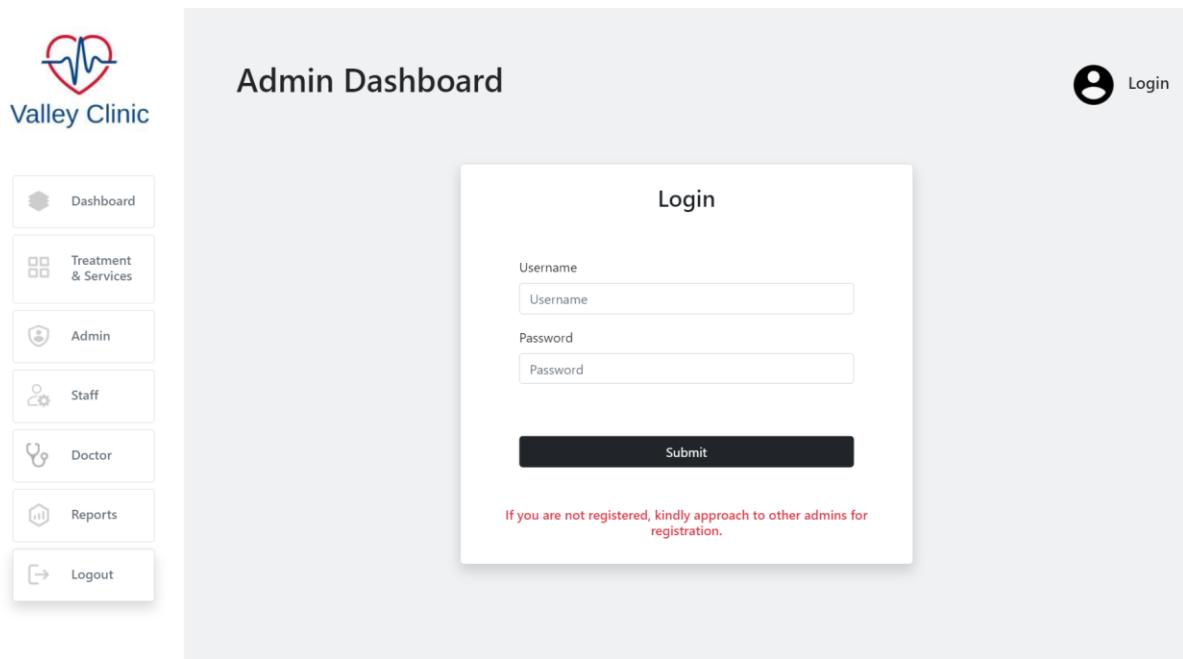


Figure 33: Admin Dashboard Login

To access the admin dashboard, admin need to login with their username and password in this login page. Admin who needs to create an account must be assisted by another admin. This will avoid anonymous user from creating an account and access the system.

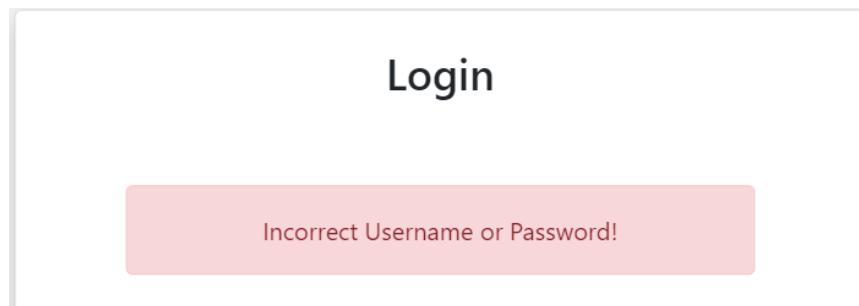


Figure 34: Admin Dashboard Login

This is the error message if admin entered the incorrect username or password.

## 15.2.2 Dashboard

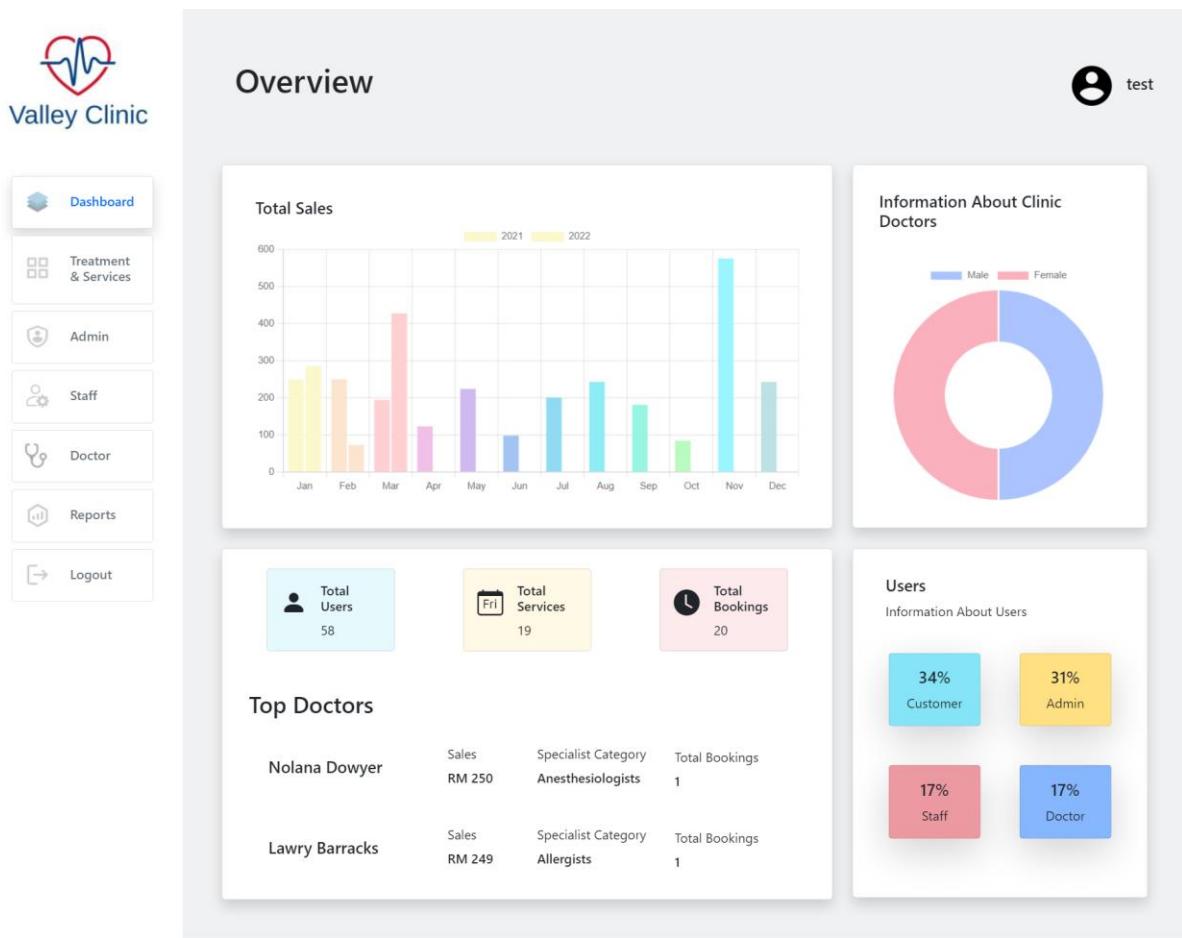


Figure 35: Admin Dashboard Overview

After admin successfully logged in, admin will be redirected to this dashboard overview page. In this page, there are a bar chart showing the total sales of year 2021 and 2022. On the top right corner, there will be the username of the admin and a doughnut chart showing percentage of doctor's gender. Below the bar chart, there is a section where total users, services and total bookings are showed. The top 2 doctors will also be listed along with their details. For the bottom right corner, this user sections shows the percentage of each type of users.

### 15.2.3 Treatment & Services

The screenshot shows the 'Treatment & Services' section of the Admin Dashboard. On the left, there's a sidebar with icons for Dashboard, Treatment & Services (selected), Admin, Staff, Doctor, Reports, and Logout. The main area has a title 'Treatment & Services' and a user icon 'test'. It includes a search bar and buttons for Copy, CSV, Excel, PDF, and Print. A table lists treatments with columns for Treatment & Services, Specialist Category, Consultation Fee Range, and actions (Update, Remove). The table shows entries like Allergy (Asthma), Audiology (Hearing), Bariatric Surgery (Weight Loss Surgery), etc. At the bottom, it says 'Showing 1 to 10 of 19 entries' and has navigation links for First, Previous, Next, and Last.

Treatment & Services	Specialist Category	Consultation Fee Range	Update	Remove
Allergy (Asthma)	Allergy	RM51 - RM80	[Update]	[Remove]
Audiology (Hearing)	Audiology	RM51 - RM80	[Update]	[Remove]
Bariatric Surgery (Weight Loss Surgery)	Bariatric Surgery	RM51 - RM80	[Update]	[Remove]
Birthing Centers	Birth	RM126 - RM235	[Update]	[Remove]
Cardiology (Heart)	Cardiology	RM126 - RM235	[Update]	[Remove]
Emergency Medicine	Emergency Medicine	RM126 - RM235	[Update]	[Remove]
Endocrinology (Diabetes & Metabolism)	Endocrinology	RM30 - RM50	[Update]	[Remove]
Massage Therapy	Massage Therapy	RM81 - RM125	[Update]	[Remove]
Nephrology (Kidneys)	Nephrology	Please Select A Range	[Update]	[Remove]
Oncology (Cancer)	Oncology	RM126 - RM235	[Update]	[Remove]

Figure 36: Admin Dashboard Treatment & Services

To view all the treatment and services, admin can navigate to the menu bar on the left panel and click on the “Treatment & Services”. In this page, admin gets to search for the treatment, and view the specialist category and ratings. Furthermore, admin can also update or remove the treatment and services by clicking on the respective button. Moreover, this page allows admin to export search results with the buttons of “Copy”, “CSV”, “Excel”, “PDF”, and “Print”. Other than that, admin can also click on the row of treatment and services to view all the details. Lastly, there will be a “New Treatment & Services” button for the admin to insert new treatment and services into the system.

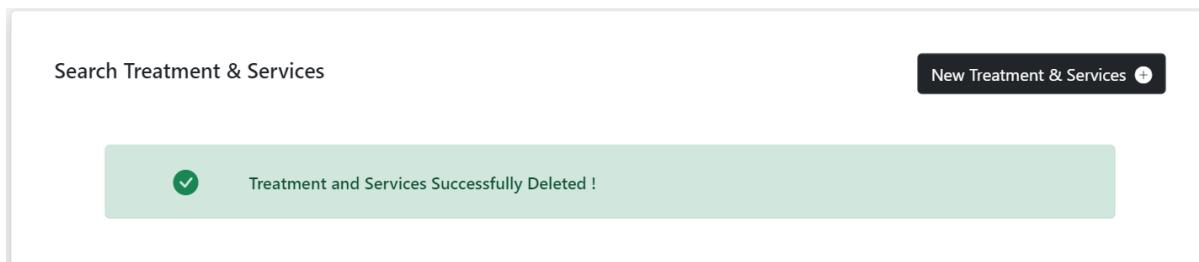


Figure 37: Admin Dashboard Delete Treatment & Services

This is the successful message when the treatment and services has been deleted.

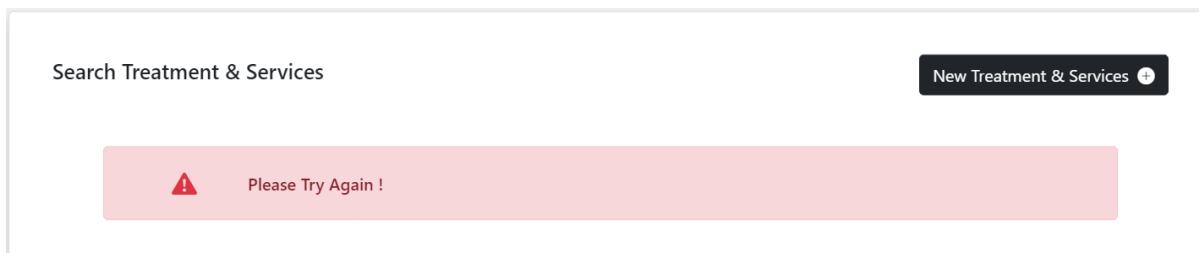


Figure 38: Admin Dashboard Delete Treatment & Services

This is the error message there is issue in deleting treatment and services.



## Treatment & Services

test

- Dashboard
- Treatment & Services
- Admin
- Staff
- Doctor
- Reports
- Logout

### New Treatment & Services

Title

Description

Specialist Category

Consultation Fee Range

Please Select A Range

Picture

Choose File No file chosen

Figure 39: Admin Dashboard New Treatment & Services

To insert a new treatment and services, admin needs to navigate to the top right corner of the treatment and services page and click on the button. All input fields are compulsory to ensure database stores sufficient information. After that, admin can fill up this form and submit. The new treatment and services will be updated.

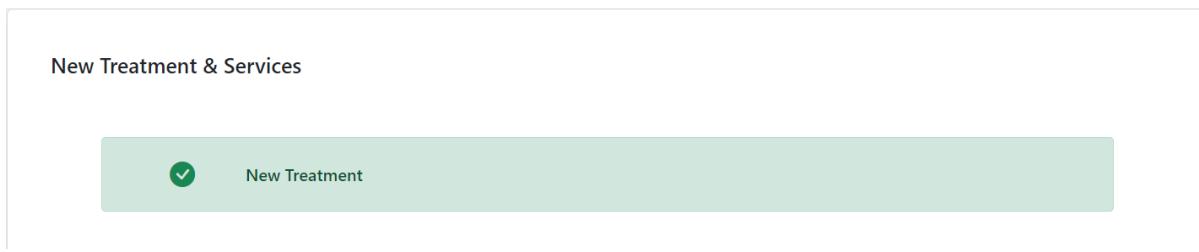


Figure 40: Admin Dashboard New Treatment & Services

This is the successful message when a new treatment is created.

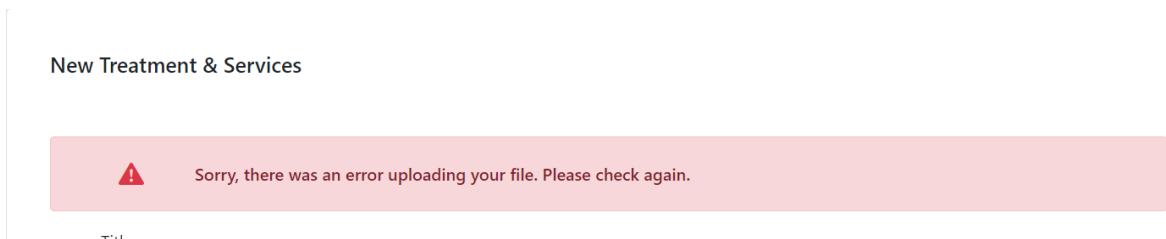


Figure 41: Admin Dashboard New Treatment & Services

This is the error message when an unsupported file type is uploaded, and the file upload is failed.



## Treatment & Services

test

- [Dashboard](#)
- [Treatment & Services](#)
- [Admin](#)
- [Staff](#)
- [Doctor](#)
- [Reports](#)
- [Logout](#)

### Update Treatment & Services



#### Title

Allergy (Asthma)

#### Description

Care for those with allergies, asthma and immunodeficiency. We strive to control your symptoms and improve your quality of life.

#### Specialist Category

Allergy

#### Consultation Fee Range

RM51 - RM80

Submit

Figure 42: Admin Dashboard Update Treatment & Services

To update treatment and services, admin can click on the “Update” button on that row of treatment and services. All input fields are compulsory to ensure database stores sufficient information. After admin done updating the information, the treatment and services will be updated accordingly.

Update Treatment & Services



Treatment and Services Successfully Updated !

*Figure 43: Admin Dashboard Update Treatment & Services*

This is the successful message when the treatment and services has been updated.

Update Treatment & Services



Please Try Again !

*Figure 44: Admin Dashboard Update Treatment & Services*

This is the error message when the treatment and services fail to update.



## Treatment & Services

test

- Dashboard
- Treatment & Services
- Admin
- Staff
- Doctor
- Reports
- Logout

### Cardiology (Heart)

Show your heart some love with personalized care that meets your exact needs. Mayo Clinic Health System is here to help you and your family prevent heart-based complications and provide lifesaving treatment when you need it most. Our cardiologists and heart care experts offer a range of services and clinics so that your care plan can be just as unique as you are. We bring the power of collaboration and the life-changing benefits of Mayo Clinic close to home. It's heart care designed for you.

#### Consultation Fee Range

RM126 - RM235

#### Specialist Category

Cardiology



SEARCH



UPDATE



DELETE



Figure 45: Admin Dashboard View Treatment & Services

Admin can view all the details of a particular treatment and services after they click on the row of treatment and services. There will also be search, update, and remove functions provided to the admin. These buttons will lead admin to their respective pages.

#### 15.2.4 Manage Administrator

The screenshot shows the 'Administrator' page of the Valley Clinic software. At the top right, there is a user icon labeled 'test'. On the left, a vertical sidebar menu includes 'Dashboard', 'Treatment & Services', 'Admin' (which is selected and highlighted in blue), 'Staff', 'Doctor', 'Reports', and 'Logout'. The main content area is titled 'Administrator' and contains a table titled 'Search Administrator Accounts'. The table has columns for 'Name' and 'Remove'. It lists 10 entries from a total of 18: agoblei, akensall6, bcriging, chundell7, gbalmeh, hclutheramb, ileupold, jbartolomivis5, lcattonnetf, and mhingeleyj. Each entry has a 'Remove' button to its right. At the bottom of the table, there are links for 'First', 'Previous', 'Next', and 'Last'. Above the table, there are buttons for 'Copy', 'CSV', 'Excel', 'PDF', and 'Print', along with a 'Search:' input field. A 'New Account' button is located at the top right of the table area.

Name	Remove
agoblei	Remove
akensall6	Remove
bcriging	Remove
chundell7	Remove
gbalmeh	Remove
hclutheramb	Remove
ileupold	Remove
jbartolomivis5	Remove
lcattonnetf	Remove
mhingeleyj	Remove

Figure 46: Admin Dashboard Search Administrator Account

To view all the staff accounts, admin can navigate to the menu bar on the left panel and click on the “Admin”. In this page, admin gets to search for the admin. Furthermore, admin can also remove the admin account by clicking on the remove button. Moreover, this page allows admin to export search results with the buttons of “Copy”, “CSV”, “Excel”, “PDF”, and “Print”. Lastly, there will be a “New Account” button for the admin to insert new admin into the system.



Figure 47: Admin Dashboard Delete Administrator Account

This is the error message when the fail to delete an admin account.

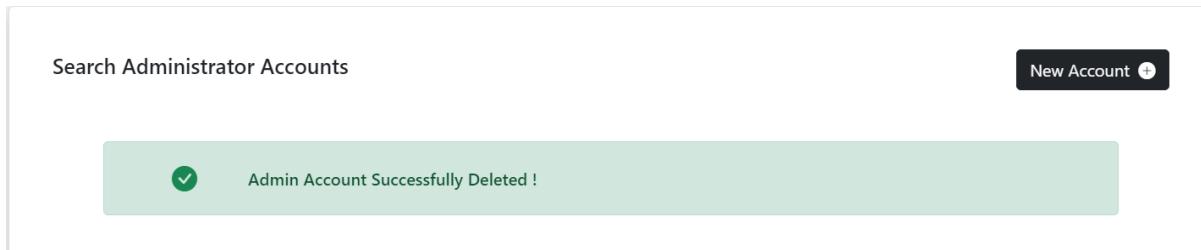


Figure 48: Admin Dashboard Delete Administrator Account

This is the successful message when the admin account has been deleted.

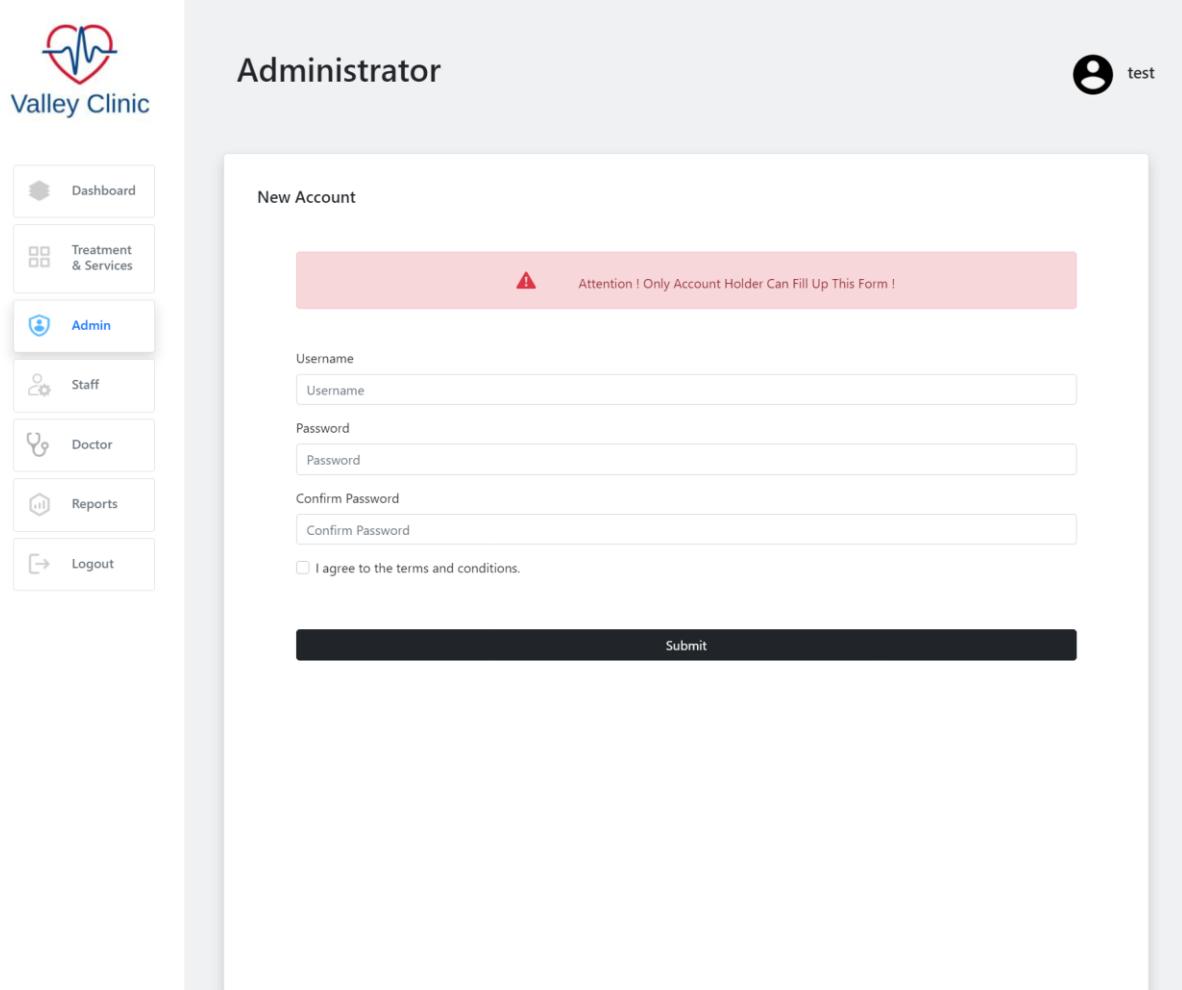


Figure 49: Admin Dashboard New Administrator Account

After admin click on the “New Account” button, this page will be displayed. To avoid other anonymous user simply register an account in the system, admin need assistance from another administrator to create a new account. The username and password can only be input by the account holder. All input fields are compulsory to ensure database stores sufficient information. After admin submits the form, there will be message to inform the status of creating a new account.

### New Account

 Username already taken! Please choose another username.  
Attention ! Only Account Holder Can Fill Up This Form !

Figure 50: Admin Dashboard New Administrator Account

This is the error message when the username already taken.

### New Account

 Please ensure your password and confirm password are matched.  
Attention ! Only Account Holder Can Fill Up This Form !

Figure 51: Admin Dashboard New Administrator Account

This is the error message when the password and confirm password does not match.

### New Account

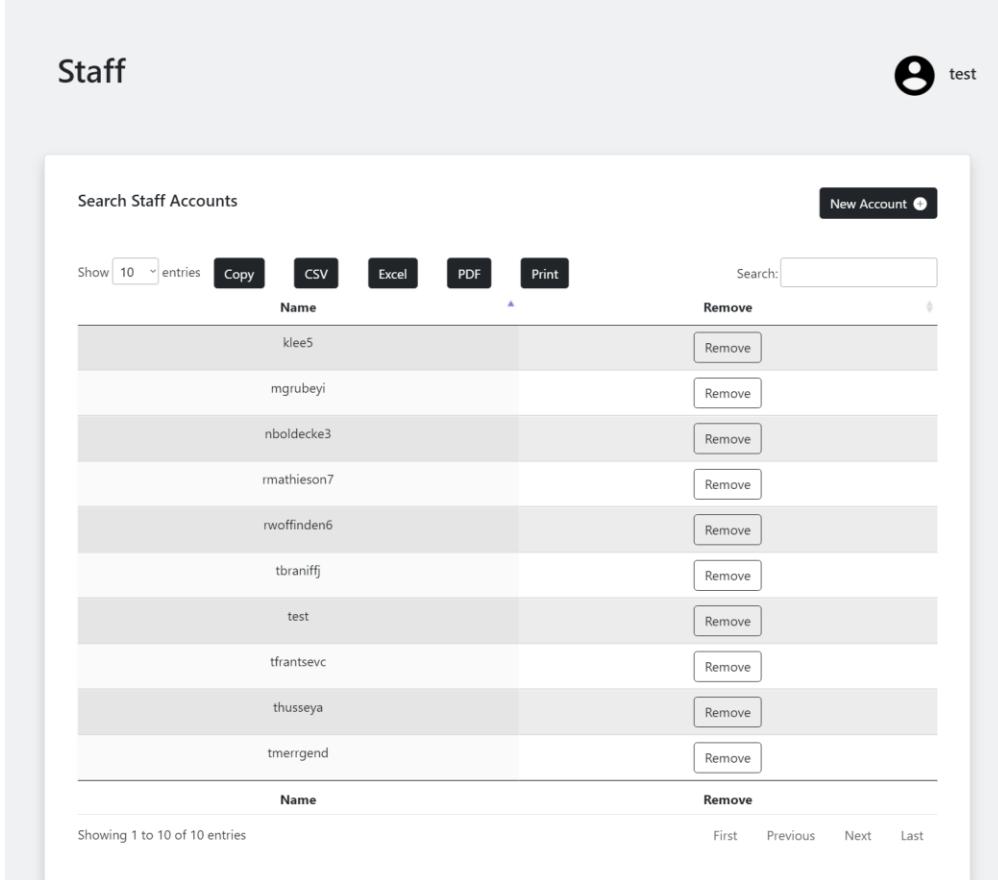


Admin Account Successfully Created !

Figure 52: Admin Dashboard New Administrator Account

This is the successful message when the admin account has been created.

### 15.2.5 Manage Staff



The screenshot shows the 'Staff' section of the Admin Dashboard. At the top, there's a logo for 'Valley Clinic' featuring a heart with an ECG line. On the right, a user icon labeled 'test' is visible. A sidebar on the left contains links for 'Dashboard', 'Treatment & Services', 'Admin', 'Staff' (which is highlighted in blue), 'Doctor', 'Reports', and 'Logout'. The main content area is titled 'Staff' and contains a table titled 'Search Staff Accounts'. The table has two columns: 'Name' and 'Remove'. The names listed are: klee5, mgrubeyi, nholdecke3, rmathieson7, rwoffinden6, tbraniffj, test, frantsevc, thusseya, and tmerrgend. Each name has a 'Remove' button to its right. Above the table, there are buttons for 'Copy', 'CSV', 'Excel', 'PDF', and 'Print'. A search bar is also present. At the bottom, it says 'Showing 1 to 10 of 10 entries' and includes navigation buttons for 'First', 'Previous', 'Next', and 'Last'.

Figure 53: Admin Dashboard Search Staff Account

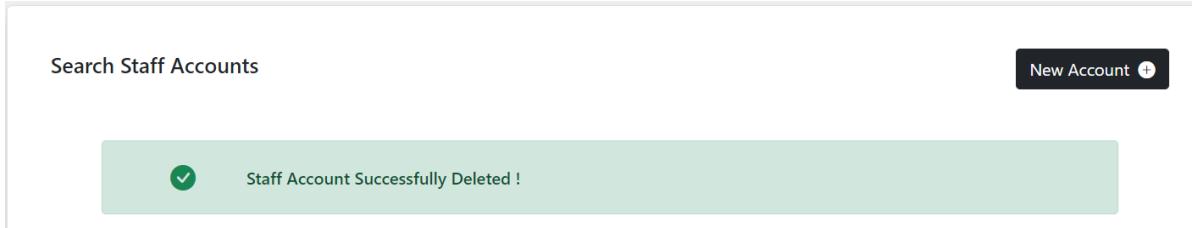
To view all the staff accounts, admin can navigate to the menu bar on the left panel and click on the “Staff”. In this page, admin gets to search for the staff. Furthermore, admin can also remove the staff account by clicking on the respective button. Moreover, this page allows admin to export search results with the buttons of “Copy”, “CSV”, “Excel”, “PDF”, and “Print”. Lastly, there will be a “New Account” button for the admin to insert new staff into the system.



This screenshot shows the same 'Search Staff Accounts' interface as Figure 53. However, a red error message box is displayed at the bottom, containing a red exclamation mark icon and the text 'Please Try Again !'.

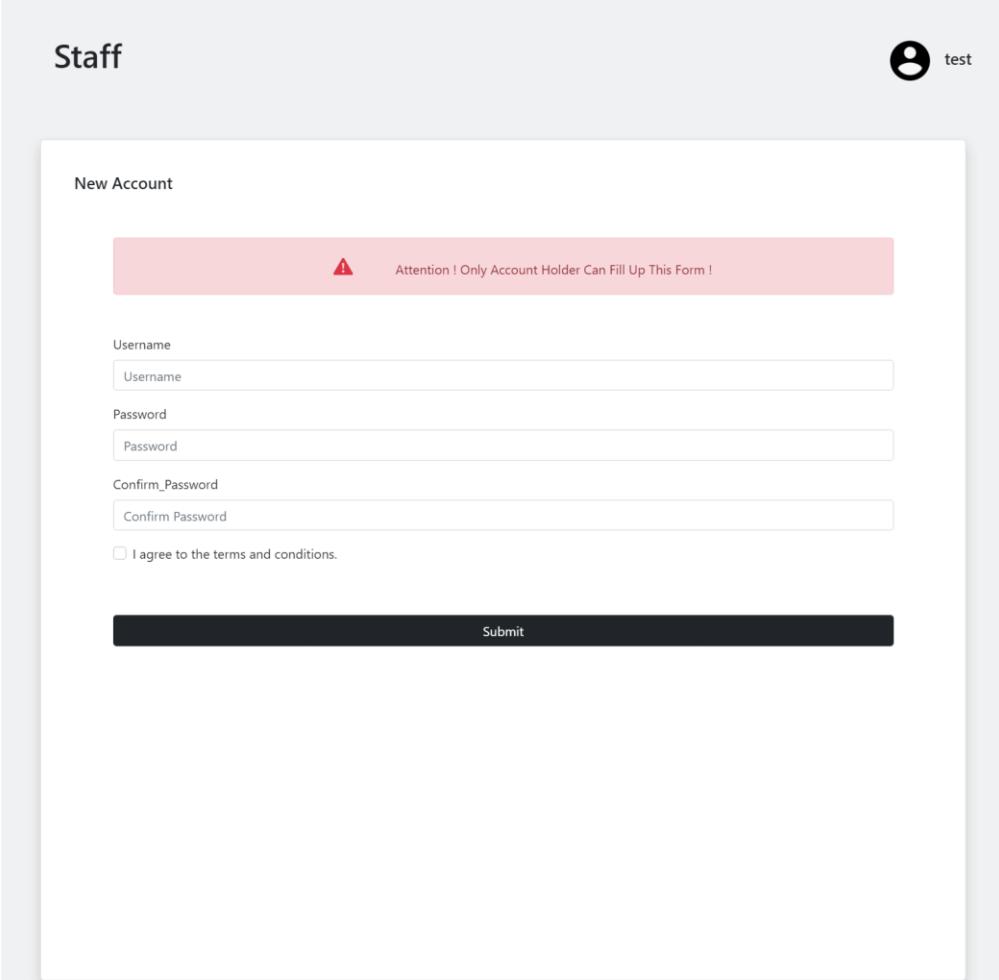
Figure 54: Admin Dashboard Delete Staff Account

This is the error message when fail to delete the staff account.



*Figure 55: Admin Dashboard Delete Staff Account*

This is the successful message when the staff account has been deleted.



The screenshot shows the 'Staff' section of the Admin Dashboard. On the left is a sidebar with a 'Valley Clinic' logo at the top, followed by a list of menu items: 'Dashboard', 'Treatment & Services', 'Admin', 'Staff' (which is highlighted in blue), 'Doctor', 'Reports', and 'Logout'. The main content area has a title 'New Account' and a red warning message box containing the text 'Attention ! Only Account Holder Can Fill Up This Form !' with an exclamation mark icon. Below the message are four input fields: 'Username' (placeholder 'Username'), 'Password' (placeholder 'Password'), 'Confirm\_Password' (placeholder 'Confirm Password'), and a checkbox labeled 'I agree to the terms and conditions.' At the bottom is a large black 'Submit' button.

Figure 56: Admin Dashboard New Staff Account

After admin click on the “New Account” button, this page will be displayed. To avoid other anonymous user simply register an account in the system, staff need assistance from the administrator to create a new account. The username and password can only be input by the account holder. All input fields are compulsory to ensure database stores sufficient information. After admin submit the form, there will be message to inform whether the staff account is created.

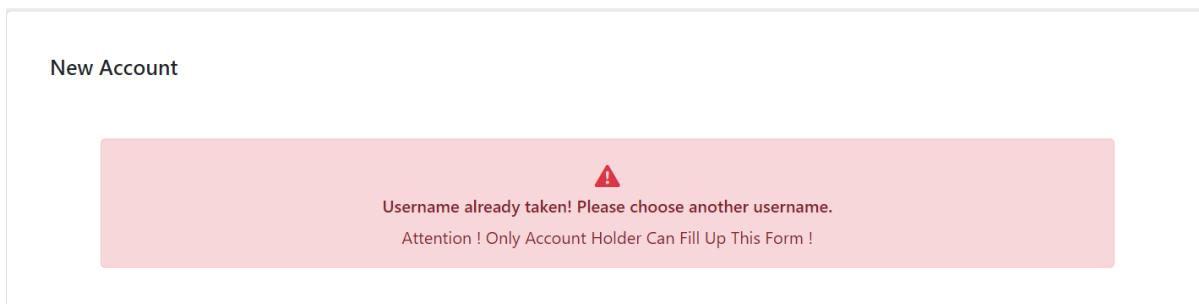


Figure 57: Admin Dashboard New Staff Account

This is the error message when the username already taken.

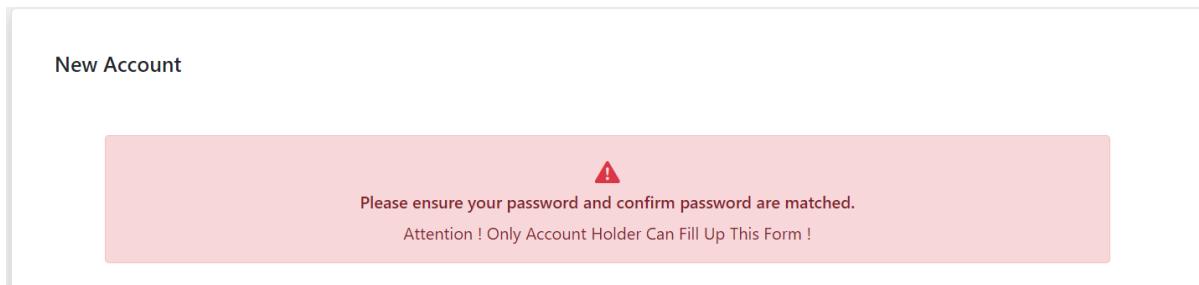


Figure 58: Admin Dashboard New Staff Account

This is the error message when the password and confirm password does not match.



Figure 59: Admin Dashboard New Staff Account

This is the successful message when the staff account has been created.

## 15.2.6 Manage Doctor

The screenshot shows the 'Doctor' page of the Admin Dashboard for Valley Clinic. The left sidebar has a 'Doctor' icon selected. The main area displays a table of doctor accounts with columns: Name, Specialist Category, Contact Number, and Remove button. Buttons at the top allow for Copy, CSV, Excel, PDF, and Print actions, along with a 'Search:' input field and a 'New Account' button.

Name	Specialist Category	Contact Number	Remove
Isador Hebblethwaite	Cardiology	619-475-8662	<button>Remove</button>
Cesare Wafer	Anesthesiologists	724-162-1565	<button>Remove</button>
Errick Mattinson	Physician	682-113-9213	<button>Remove</button>
Kris Northcliffe	Anesthesiologists	743-843-2665	<button>Remove</button>
Lawry Barracks	Allergists	176-239-2675	<button>Remove</button>
Lissa Cohani	Cardiology	961-734-6736	<button>Remove</button>
Lorrayne Olle	Cardiologists	206-839-8461	<button>Remove</button>
Marga Maruszewski	Allergy	139-203-4096	<button>Remove</button>
Nolana Dowyer	Anesthesiologists	719-205-4706	<button>Remove</button>
Randolf Selbach	Physician	938-202-5606	<button>Remove</button>

Showing 1 to 10 of 10 entries      First    Previous    Next    Last

Figure 60: Admin Dashboard Search Doctor Account

To view all the doctor accounts, admin can navigate to the menu bar on the left panel and click on the “Doctor”. In this page, admin gets to search for the doctor, and view the specialist category and contact number. Furthermore, admin can also remove the doctor account by clicking on the respective button. Moreover, this page allows admin to export search results with the buttons of “Copy”, “CSV”, “Excel”, “PDF”, and “Print”. Other than that, admin can also click on the row of doctor to view all the details. Lastly, there will be a “New Account” button for the admin to insert new doctor into the system.

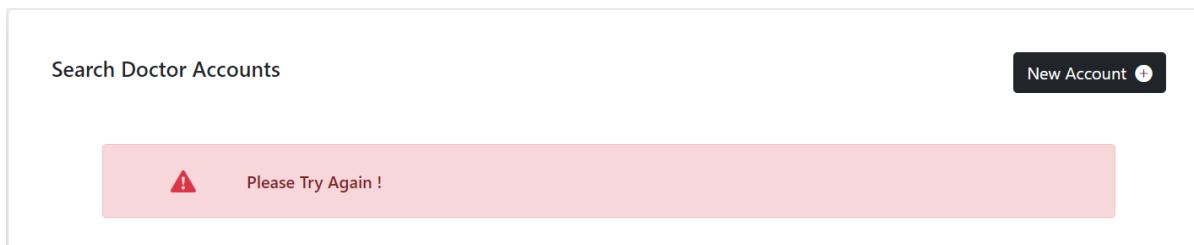


Figure 61: Admin Dashboard Delete Doctor Account

This is the error message when fail to delete the doctor account.

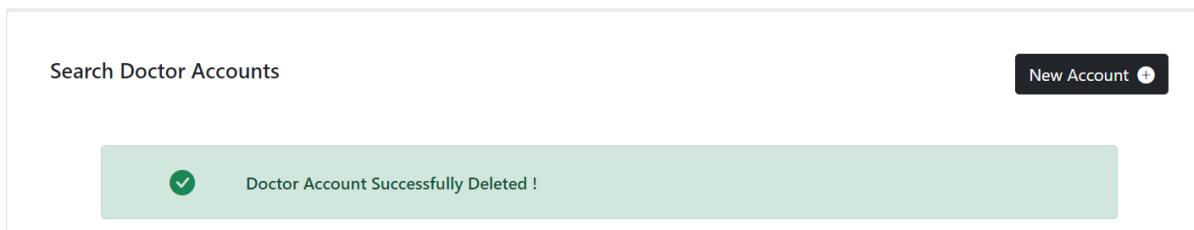


Figure 62: Admin Dashboard Delete Doctor Account

This is the successful message when the doctor account has been deleted.

The screenshot shows the Admin Dashboard of the Valley Clinic system. On the left, there is a vertical sidebar with a navigation menu:

- Dashboard
- Treatment & Services
- Admin
- Staff
- Doctor** (highlighted)
- Reports
- Logout

The main content area is titled "Doctor". At the top right, there is a user profile icon labeled "test". The central part of the screen displays a "New Account" form. A red warning message at the top right of the form area says "Attention ! Only Account Holder Can Fill Up This Form !". The form fields are as follows:

- Username: An input field labeled "Username".
- Password: An input field labeled "Password".
- Confirm\_Password: An input field labeled "Confirm Password".
- A checkbox labeled "I agree to the terms and conditions.".
- A large black "Submit" button at the bottom.

Figure 63: Admin Dashboard New Doctor Account

After admin click on the “New Account” button, this page will be displayed. To avoid other anonymous user simply register an account in the system, doctor need assistance from the administrator to create a new account. The username and password can only be input by the account holder. All input fields are compulsory to ensure database stores sufficient information. After admin submit the form, there will be message to inform whether the doctor account is created.

New Account

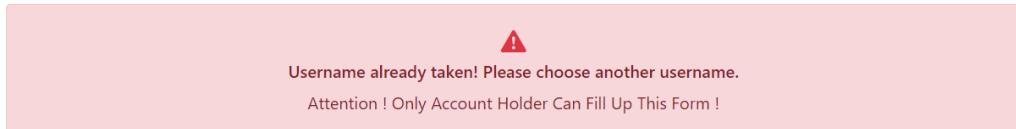


Figure 64: Admin Dashboard New Doctor Account

This is the error message when the username already taken.

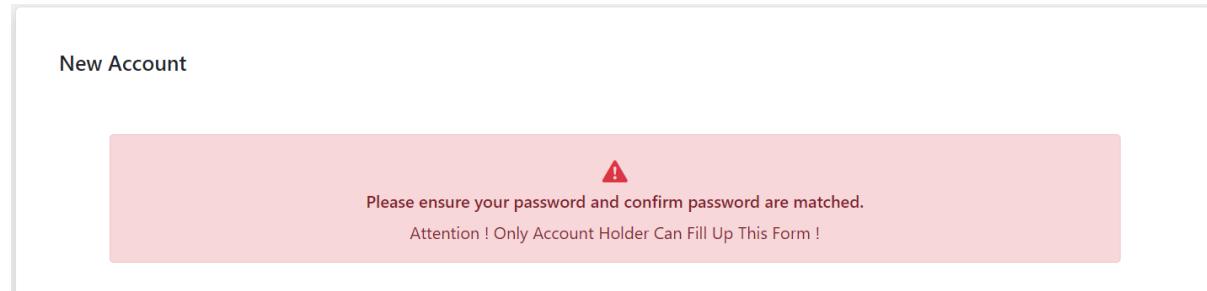


Figure 65: Admin Dashboard New Doctor Account

This is the error message when the password and confirm password does not match.

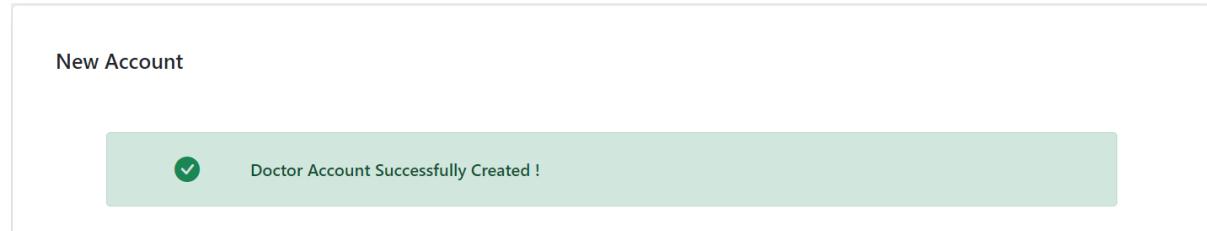


Figure 66: Admin Dashboard New Doctor Account

This is the successful message when the doctor account has been created.

The screenshot shows the Admin Dashboard interface for a medical clinic named 'Valley Clinic'. On the left, there is a vertical sidebar with navigation links: Dashboard, Treatment & Services, Admin, Staff, Doctor (which is currently selected and highlighted in blue), Reports, and Logout. The main content area has a title 'Doctor' and a user profile picture labeled 'test'. Below the profile, the following details are displayed:

Attribute	Value
Name	Erick Mattinson
Username	Errick
Specialist Category	Physician
Gender	Female
Date Of Birth	1982-06-28
Contact Number	682-113-9213
Email	emattinson3@abc.net.au
Address	767 Sauthoff Plaza

On the right side of the details, there are two buttons: a 'SEARCH' button with a magnifying glass icon and a 'DELETE' button with a trash can icon.

Figure 67: Admin Dashboard View Doctor Account

Admin can view all the details of a particular doctor after admin click on the row of doctor account. Doctor's name, username, specialist category, gender, date of birth, contact number, email and address will be shown. There will also be search and remove functions provided to the admin. Admin can go to these pages the access the functions.

## 15.2.7 Reports

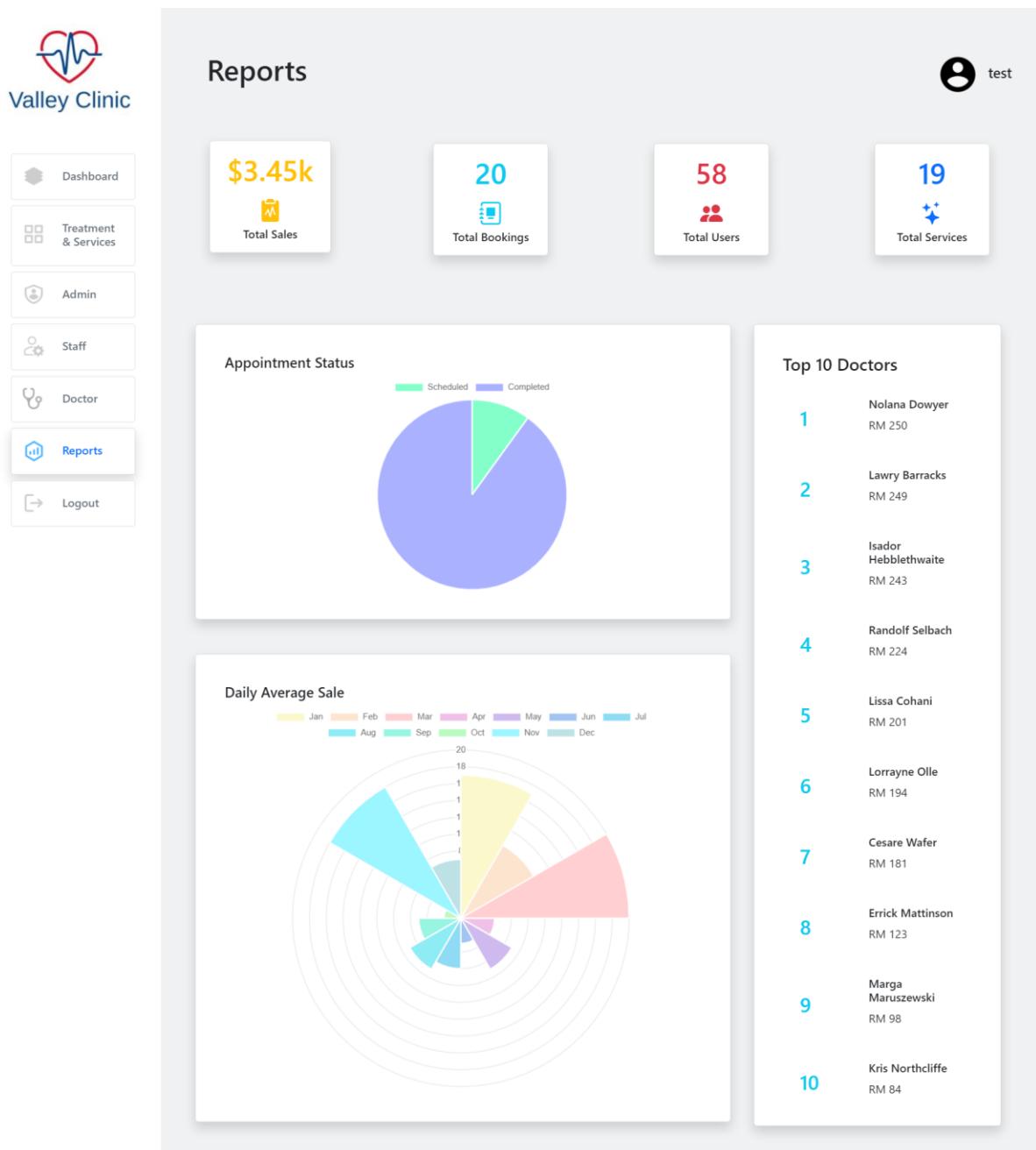


Figure 68: Admin Dashboard Report

To view the reports, admin can navigate to the menu bar on the left panel and click on the “Reports”. In this page, admin gets to view the total sales, total bookings, total users, and total services. Furthermore, there will be pie chart of appointment status. Below this pie chart, there will be a polar area chart representing the daily average sale for each month. Lastly, the top 10 doctors will be displayed along with their sales.

## 15.2.8 Logout

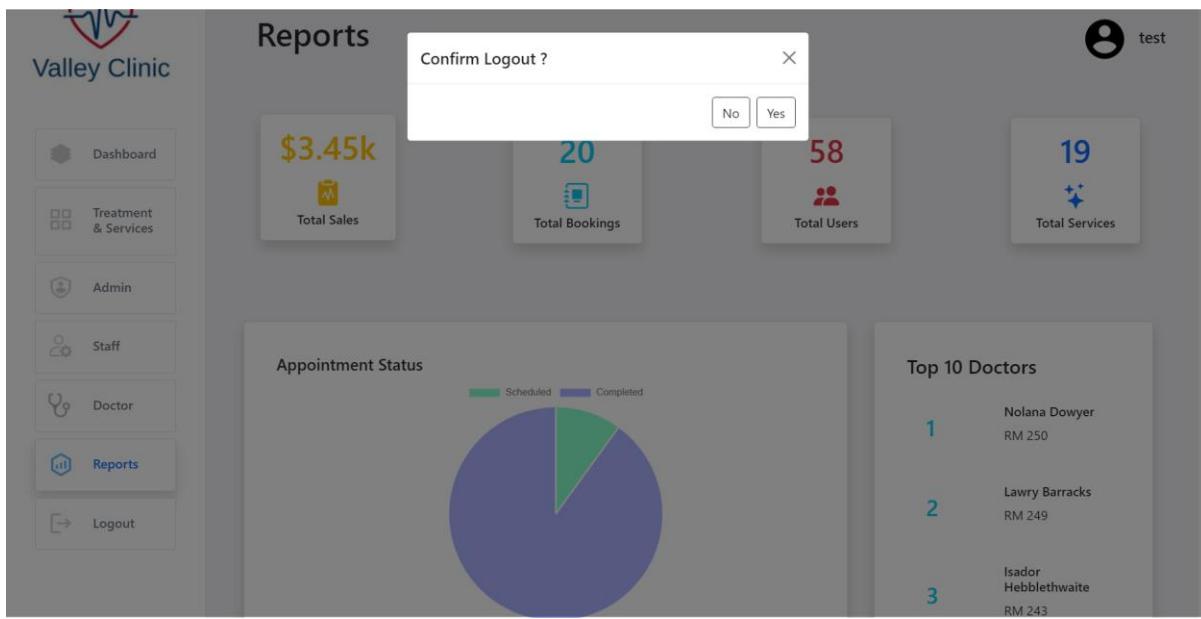


Figure 69: Admin Dashboard Logout

This is the logout message appears when admin click the “Logout” at the left panel. If admin wish to logout, by clicking yes, the admin will be logged out successfully. While admin regrets and wish to continue using admin dashboard, the logout message will be dismissed and resume in the current page.

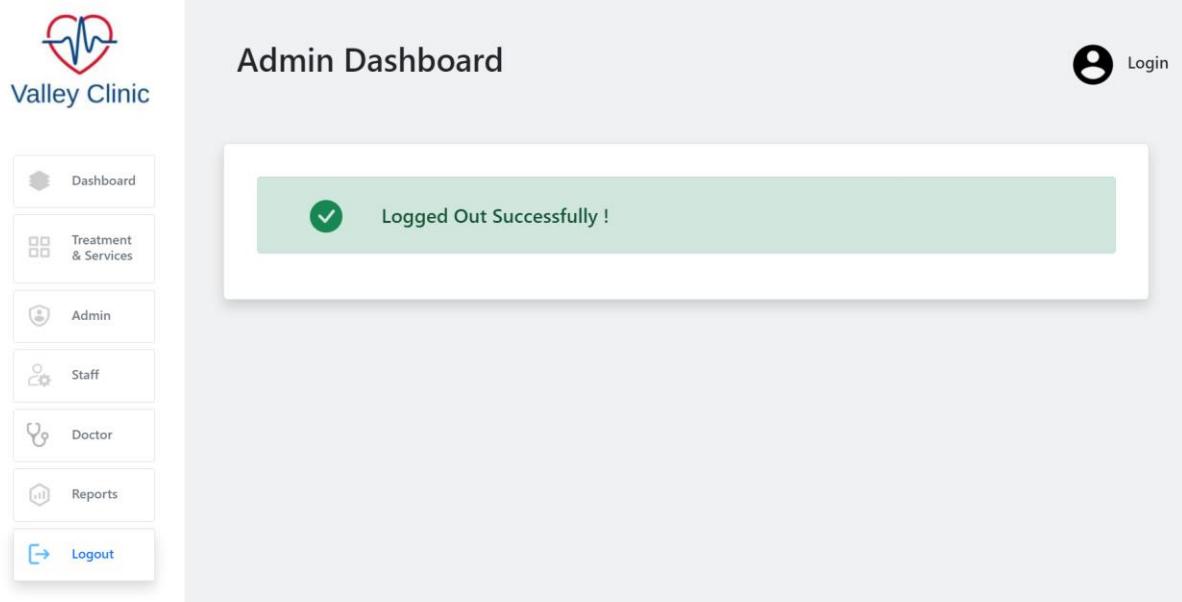


Figure 70: Admin Dashboard Logout

This is the successful message when the admin has logged out the system.

## 16.3 Doctor Module

### 16.3.1 Doctor's Login

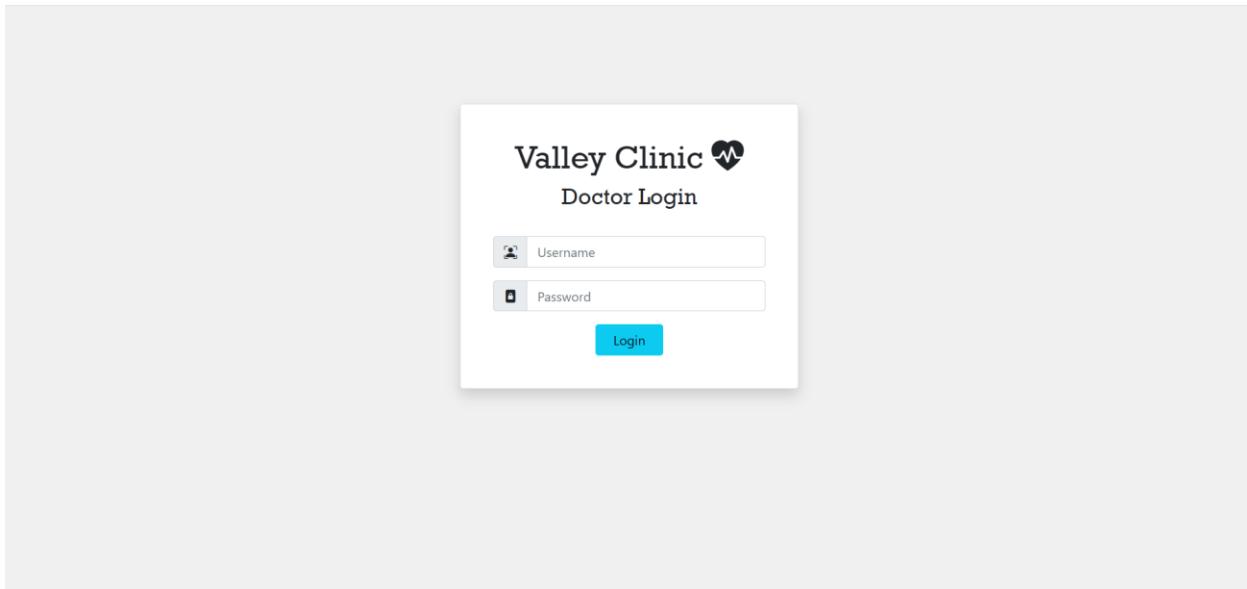


Figure 71 Doctor Login Page

To get access to the doctor module, the users will first need to login at the login page. Users need to key in their username and password to login, and the system will check whether the login criteria is correct or not.

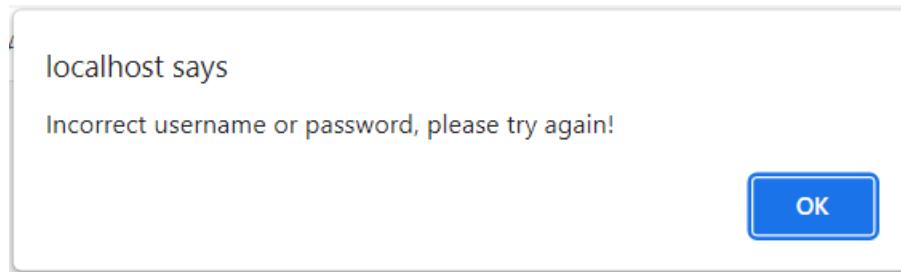


Figure 72 Doctor Login Error

If the login criteria key in by the users is incorrect an alert box will appear and notify the users that their username or password are incorrect. By clicking the 'OK' button the users will be redirected to the login page.



Figure 73 Doctor Login Success

If the users had key in the correct username and password an alert box will appear and show ‘Login Success!’. By clicking the ‘OK’ button, the users will be directed to the doctor home page. From this moment the users can get access to the full doctor module.

### 16.3.2 Doctor’s Homepage

Today's Schedule Date: 6/3/2022

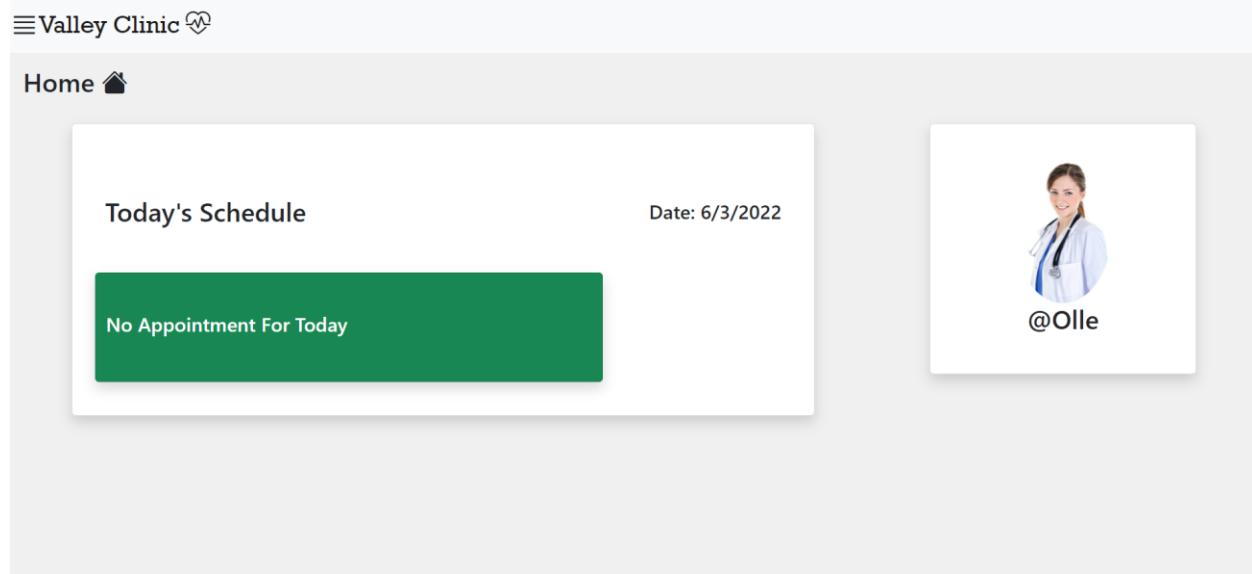
AppointmentID: 12  
PatientID : 2  
Patient Name : Roselle Wolffers  
Time : 10:00-11:00

AppointmentID: 13  
PatientID : 2  
Patient Name : Roselle Wolffers  
Time : 12:00-13:00

@LawryB

Figure 74 Doctor Homepage

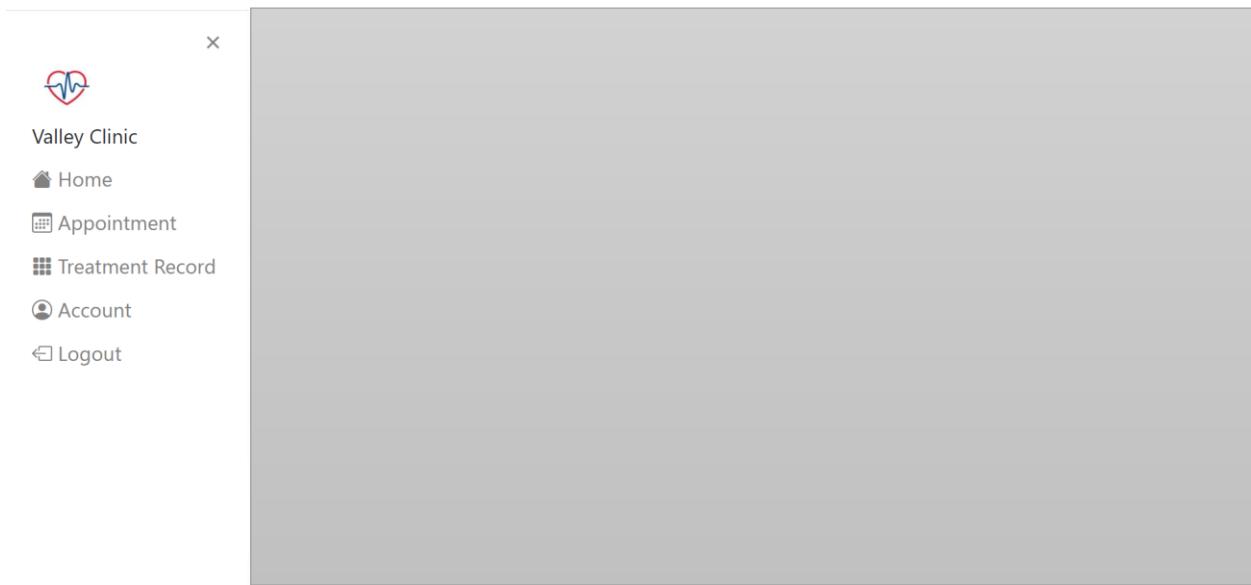
This is the doctor homepage; it is the first page where it will show when the users pass the login session. In this page, the schedule of the day of the doctor will be show. Beside the schedule it will show the user profile picture and the username.



*Figure 75 Example Doctor Homepage of Empty Schedule*

This is another example of the doctor homepage, but in this case the doctor's schedule is empty for the day. Therefore, a different output will be present to notify the doctor that they have no appointment for the day.

### 16.3.3 Side Navigation Bar



This is the navigation bar for the whole doctor module. Users can direct to each and different page through this navigation bar. Users just need to click on the appropriate button, and it will direct them to the page they would like to go.

### 16.3.3 Doctor's Appointment Page

The screenshot shows a table titled "Appointment" listing five entries. The columns are AppointmentID, PatientID, Patient Name, Time, and Date. The data is as follows:

AppointmentID	PatientID	Patient Name	Time	Date
4	2	Roselle Wolffers	12:00-13:00	2022-02-23
5	6	Angelica Lukasen	12:00-13:00	2022-02-28
7	5	Rowan Riccione	12:00-13:00	2021-02-26
8	4	Erny Scarrisbrick	10:00-11:00	2022-02-25
9	2	Roselle Wolffers	10:00-11:00	2022-02-27

Showing 1 to 5 of 5 entries

First Previous Next Last

Figure 76 Doctor Appointment Page

In this page, a table of data will be shown. The data is all about the appointment of the doctor, containing all past and future appointment. Above the data there is also a search function which

allow the users to search for their desire data. Around the table there are also some buttons that can help the users to filter the table.

#### 16.3.4 Doctor's Treatment Record Page

The screenshot shows a web-based treatment record page for a doctor. At the top left is the logo 'Valley Clinic'. Below it, the title 'Treatment Record' is displayed. A search bar labeled 'Search:' is present. On the left, a dropdown menu says 'Show 10 entries'. The main content is a table with three columns: 'AccountID', 'Name', and 'Record'. The table contains four rows with the following data:

AccountID	Name	Record
2	Roselle Wolffers	<button>View</button>
4	Erny Scarrisbrick	<button>View</button>
5	Rowan Riccione	<button>View</button>
6	Angelica Lukasen	<button>View</button>

At the bottom left, it says 'Showing 1 to 4 of 4 entries'. On the right, there are navigation buttons: 'First', 'Previous', 'Next', and 'Last'.

Figure 77 Doctor Treatment Record Page

This is the treatment record page where design to let the doctor find and view the treatment record details of their patient. At first it will show a table of account that belongs to the doctor's patient. In the table it will shows the patient's AccountID, name and a view button.

### Treatment Record-

Roselle Wolffers

Show  entries

Search:

AppointmentID	Time	Date	Details
1	10:00-11:00	2022-03-03	<input type="button" value="View"/> <input type="button" value="Edit"/>
11	10:00-11:00	2022-03-04	<input type="button" value="View"/> <input type="button" value="Edit"/>
12	10:00-11:00	2022-03-06	<input type="button" value="View"/> <input type="button" value="Edit"/>
13	12:00-13:00	2022-03-06	<input type="button" value="View"/> <input type="button" value="Edit"/>

Showing 1 to 4 of 4 entries

First Previous Next Last

Figure 78 Doctor Treatment Record Page

This is the page where the user will be directed after clicking the view button on figure 88. In this page all appointment record of the selected patient will be show in the table. In the table the data and time of the appointment will also be shown as well. User can also search for the data they want by keying in the key words. Anyway, every row of appointment will have two button which is the ‘View’ button and the ‘Edit’ button. The view button allow the users to view the treatment record notes of the appointment and the edit button will allow the users to edit the treatment record notes and save it to database.

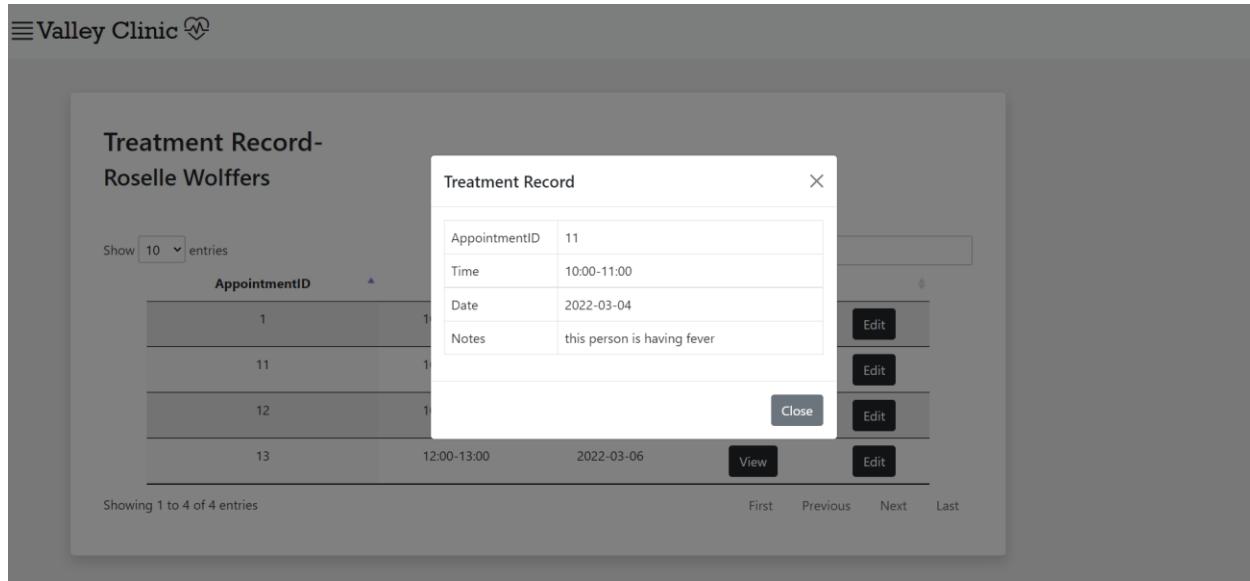


Figure 79 Doctor View Treatment Record

This is the modal which will show up after the using clicking the view button on Figure 89, the data will be show in a table form. Below the table there are a close button which enable the users to close the modal.

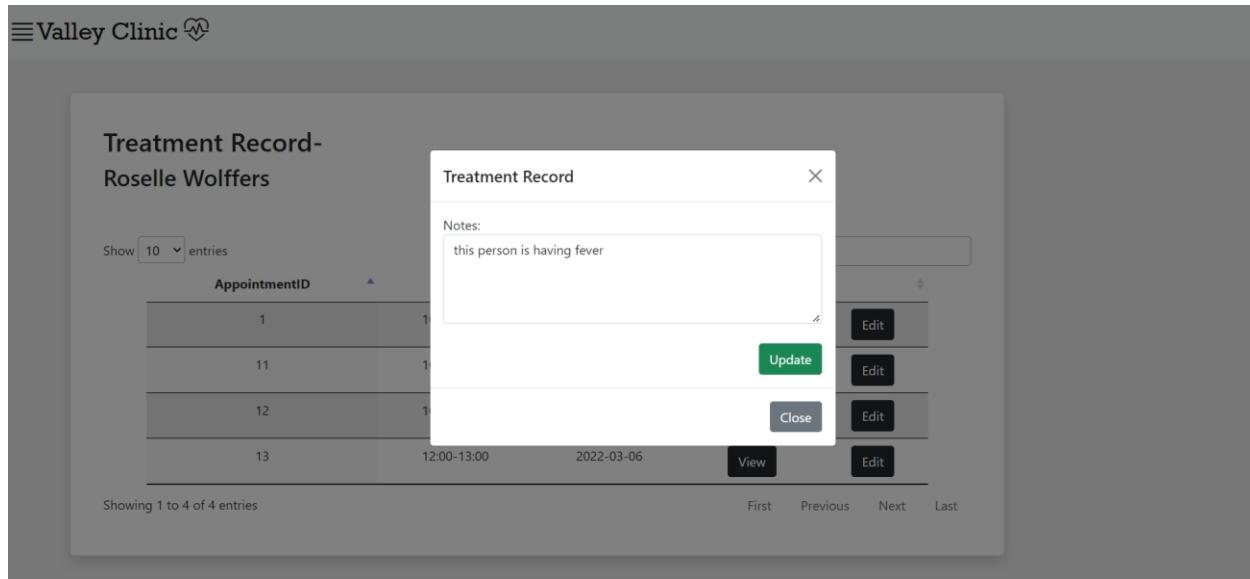


Figure 80 Doctor Edit Treatment Record

This is the modal which will appear after the user click the edit button on Figure 89, there will be an input with existing data of the database. User can edit the data and click on the green 'Update' button so that the changes there made will be save into the database. Below the 'Update' button,

there is also a cancel button which allow the users cancel the action and close the modal immediately.

### 16.3.5 Doctor's Account Page

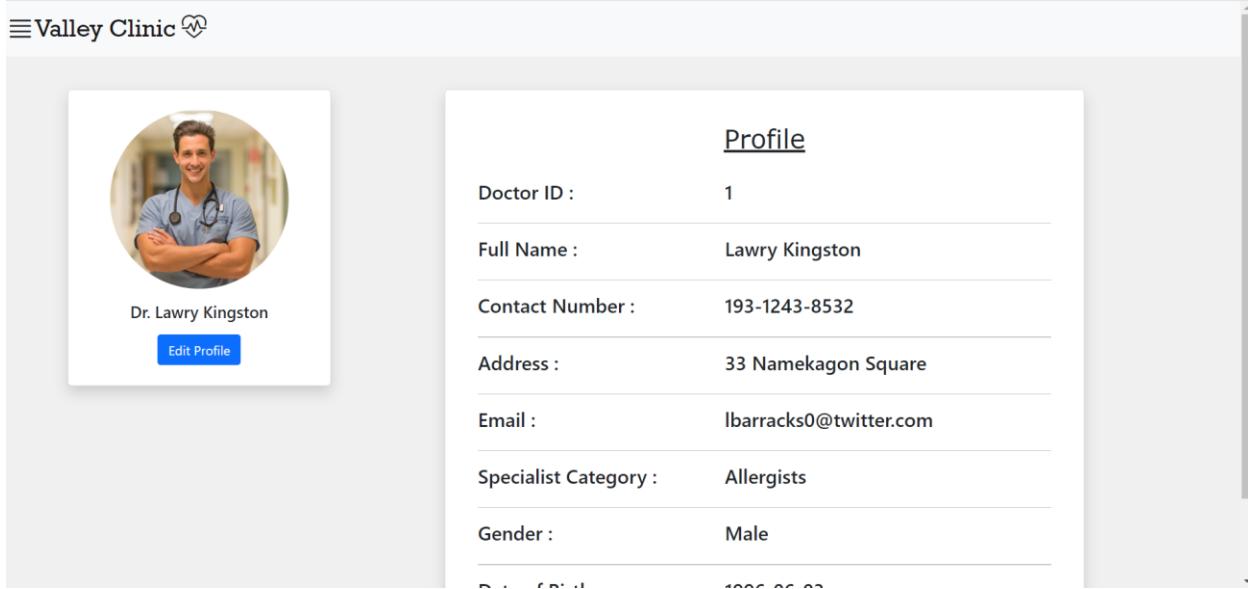


Figure 81 Doctor Account Page

This is the doctor account page where all information of the account will be shown. User can also make some edit to their profile in this page. They can edit their account details by clicking on the 'Edit Profile' button, they can also change their profile picture by hovering to the picture and click the button on it.

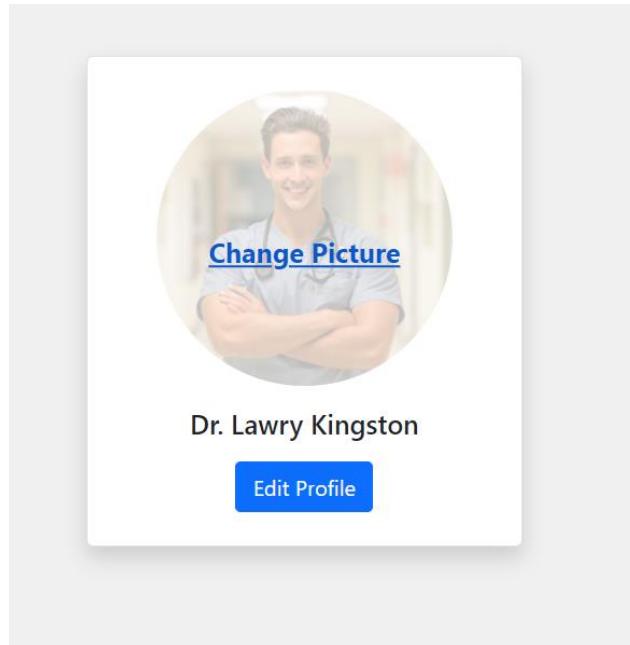


Figure 82 Change Profile Picture

When the user hover to the profile picture, a ‘Change Picture’ text will appear which enable the users to change their profile picture.

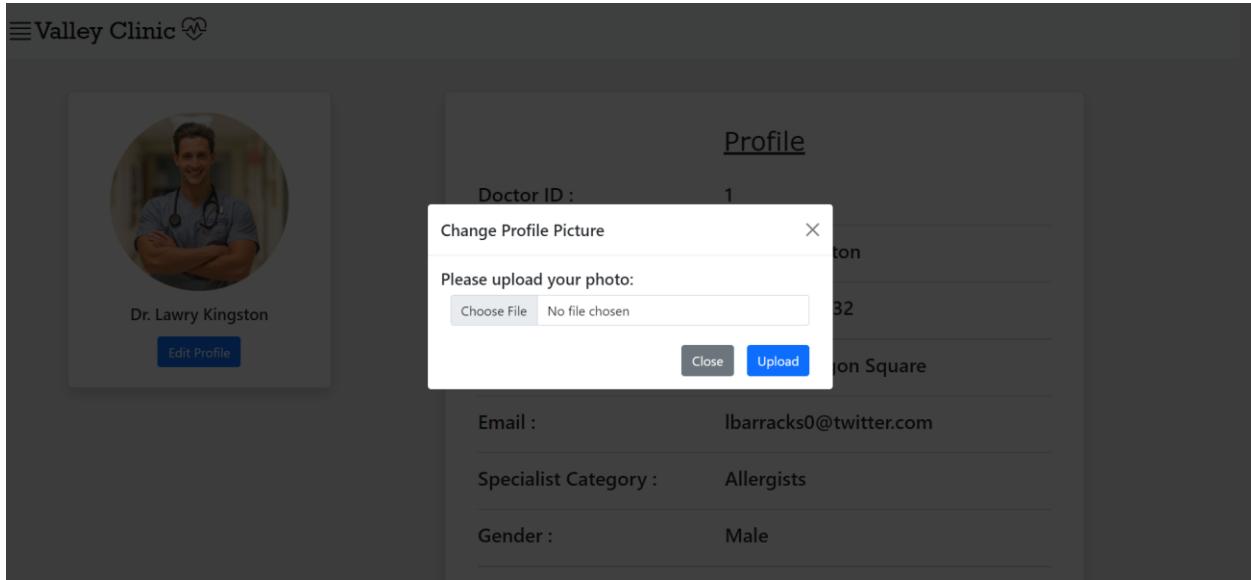


Figure 83 Upload Profile Picture

After clicking on the ‘Change Picture’ text, this modal will appear. In this modal there will be an input box accepting file data where user can upload only image files like jpg, png, png etc. only. There will be two buttons at the end of this modal which is the ‘Cancel’ button and ‘Upload’

button. The ‘Cancel’ button will close the modal immediately and the ‘Upload’ button will upload the file into the database.

The screenshot shows a modal window titled 'Edit Profile'. It contains several input fields: 'Name' (Lawry Kingston), 'Specialist Category' (Allergists), 'Contact Number' (193-1243-8532), 'Gender' (Male), 'Address' (33 Namekagon Square), 'Date of Birth' (03/06/1996), 'Email' (lbarracks0@twitter.com), and a 'Reset' button. At the bottom left is a 'Back' button, and at the bottom right are 'Reset' and 'Submit' buttons. The background of the modal has a light gray gradient.

Figure 84 Edit Profile Info

This is the page where user can edit their account information. The users can direct to this page by clicking the ‘Edit Profile’ button on Figure 92. The input box will be filled by the original value save in the database. The ‘Reset’ button reset all the value in the entry box back to its original value. The ‘Submit’ button save the changes to the database. The ‘ Back’ button will direct the user back to the account page.

### 16.3.5 Logout

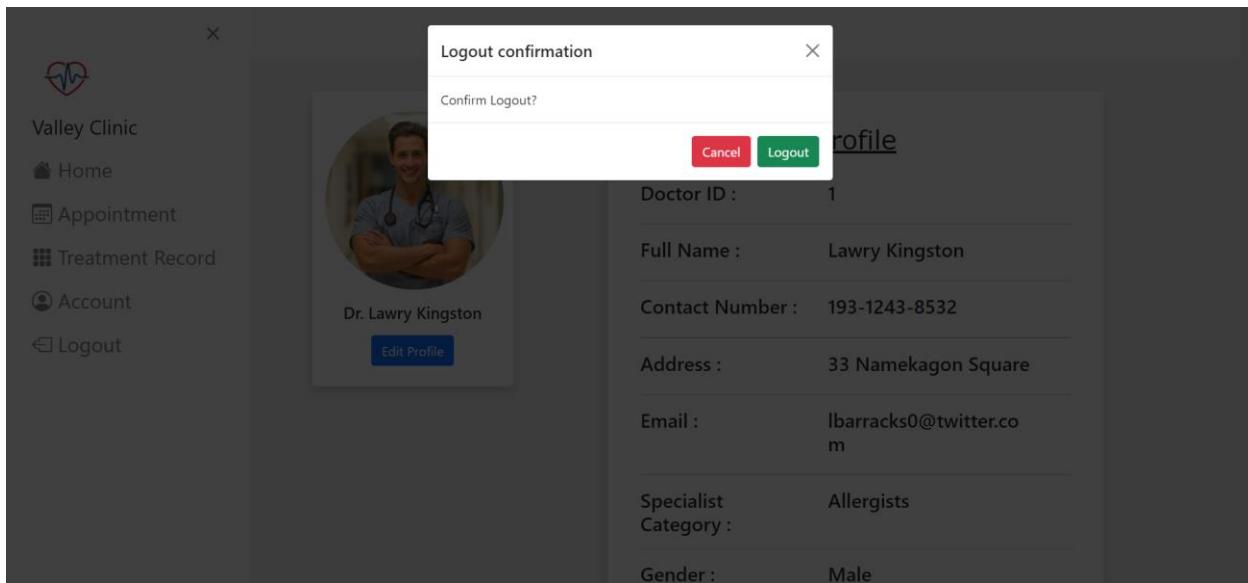


Figure 85 Logout Modal

This is the logout window when click on the logout button on the side navigation tab. It is a logout confirmation window to confirm the logout action of the users. The ‘Logout’ button will direct the users to the login page immediately and the ‘Cancel’ button will close the windows.

## 16.4 Staff Module

### 16.4.1 Staff Login

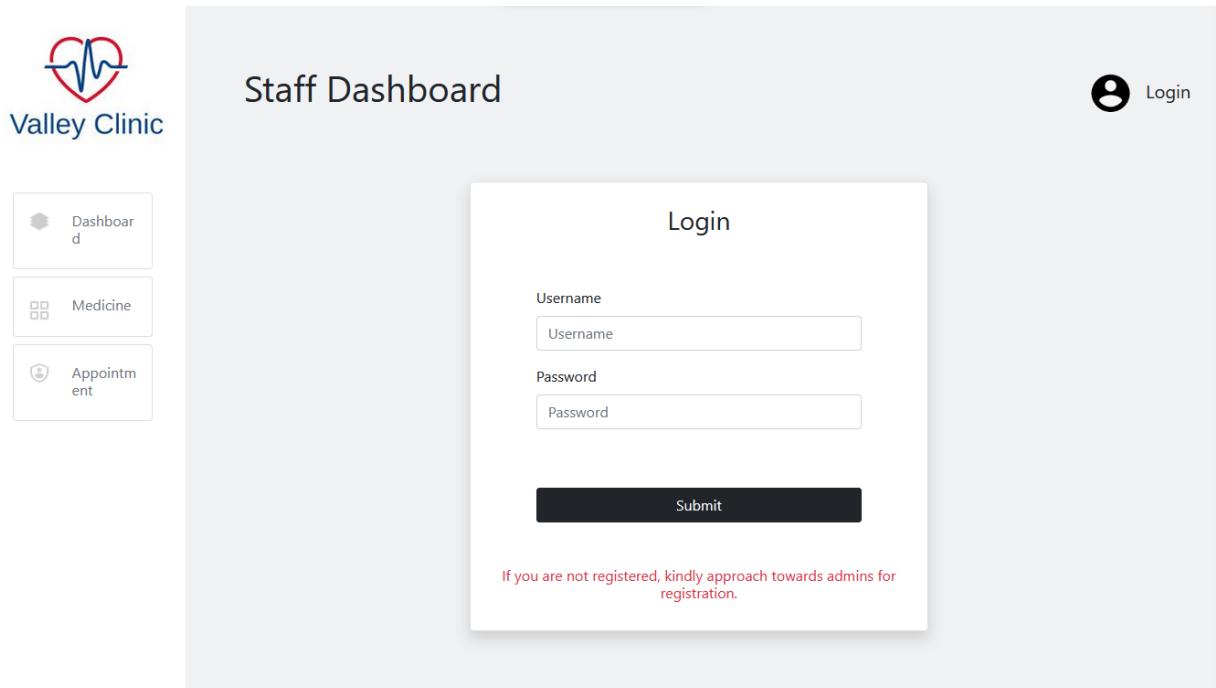


Figure 93: Staff Login Page

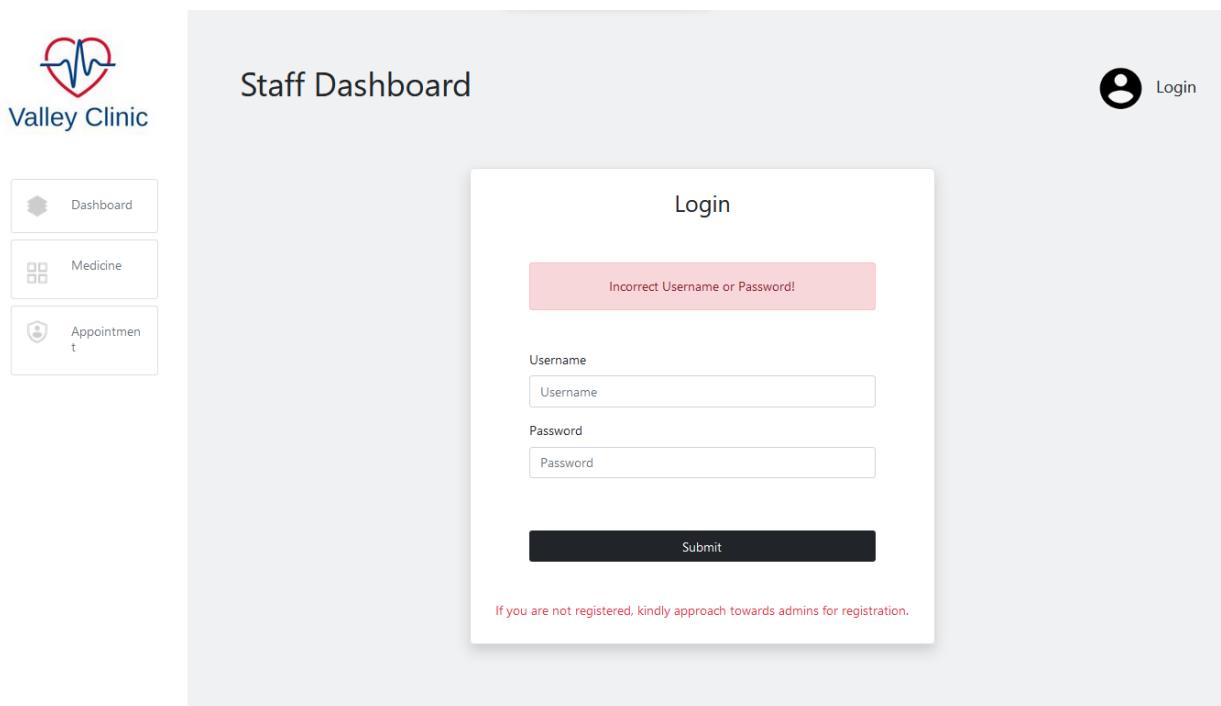


Figure 94: Incorrect Username or Password Error

To gain access into the staff dashboard, each staff are given a username and password to log into the system. If the staff does not have a registered account to log into the system, then they would need to approach the admin to have an account registered for them. On the other hand, if the username and password entered is invalid thus there will be an error message popped up onto the login page.

#### 16.4.2 Staff Overview

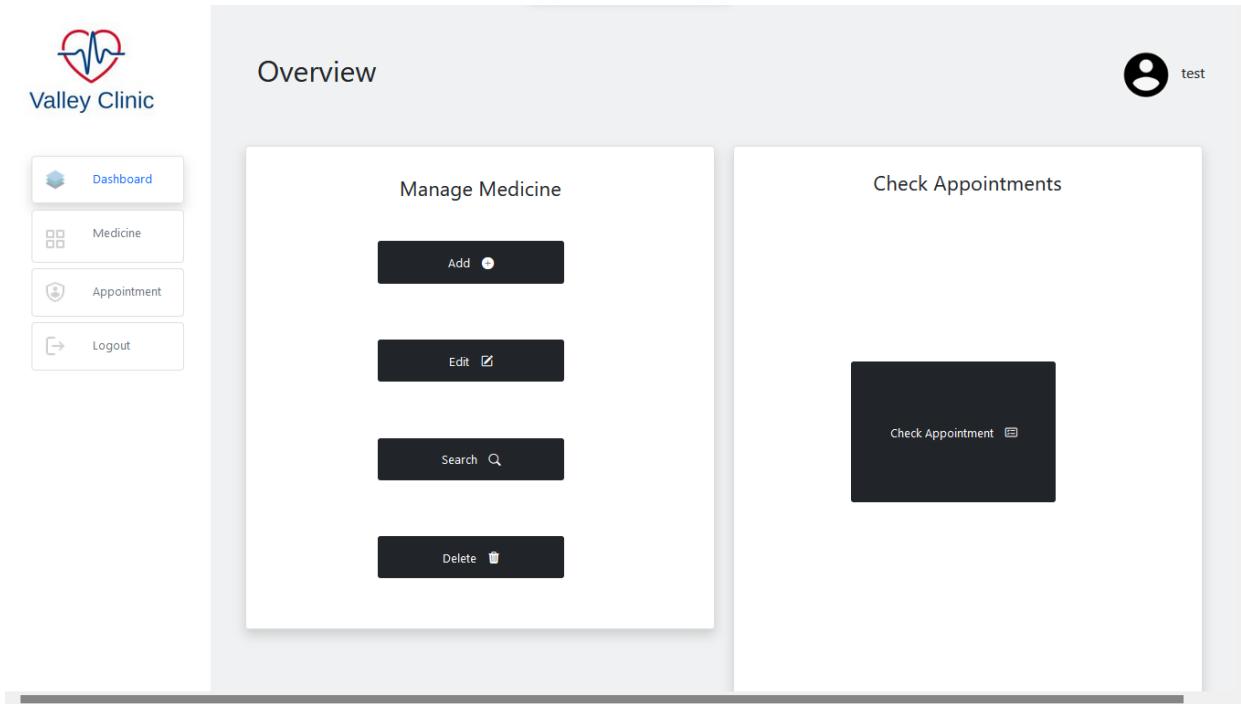


Figure 95: Staff Dashboard Overview

After inserting the username and password which will be validated, the system will bring out the dashboard page of the staff. In this page, as can be seen above on the left navigation panel, there will be 4 buttons which are dashboard, medicine, appointment, and logout. Each buttons have its own use and function. On the right side of the dashboard, it shows the 2 main functions that is available to the staff which are manage medicine and check appointments. Moreover, each button under the functions can bring the staff to the respective pages with having to click on the medicine page on the navigation panel.

### 16.4.3 Add Medicine

The screenshot shows the Valley Clinic application interface. On the left, there is a navigation sidebar with a heart icon and the text 'Valley Clinic'. Below the icon are four buttons: 'Dashboard', 'Medicine' (which is highlighted in blue), 'Appointment', and 'Logout'. The main content area has a title 'Medicine' and a sub-title 'New Medicine'. The form fields are: 'Name' (with a 'Title' input field), 'Description' (with a 'Description' input field), 'Price' (with a 'Price' input field), and 'Quantity' (with a note 'Please fill out this field.' and an input field). At the bottom are two large, dark grey buttons labeled 'Submit' and 'Clear'.

*Figure 96: Add Medicine*

After clicking on the add tab on the overview dashboard under manage medicine, it will lead to the new medicine page where the staff is able to add new medicines into the system. The required information that is needed to type in are the name, description, price, and quantity of the new medicine and click on submit which will then state new medicine added. If the staff decided to not add any new medicine after listing out the valid information then all they need to do is to clear the page and click on the dashboard button on the navigation panel.

#### 16.4.4 Edit Medicine

The screenshot shows the Valley Clinic software interface. At the top left is the Valley Clinic logo with a heart icon. At the top right is a user profile icon labeled "test". On the left is a vertical sidebar with navigation options: "Dashboard" (grey), "Medicine" (selected, highlighted in blue), "Appointment" (grey), and "Logout" (grey). The main content area has a title "Medicine" and a sub-section "Update Medicine". It contains four input fields: "Name" (Antibiotic), "Description" (Antibiotics, also known as antibacterials, are medications that destroy or slow down the growth of bacteria.), "Price" (RM123), and "Quantity" (10). A "Submit" button is at the bottom.

*Figure 97: Edit Medicine*

In the edit medicine page, as can be seen above, this is where the staff is able to edit and update the name, description, price as well as the quantity of a certain medicine. After filling out the necessary information then the staff will click on the submit button which will then update the database of the system.

### 16.4.5 Search Medicine

The screenshot shows the Valley Clinic software interface. At the top left is the Valley Clinic logo with a heart icon. The main title "Medicine" is centered above a search form. The search form includes fields for "Show 10 entries", "Search" (with placeholder "Search:"), and buttons for "Copy", "CSV", "Excel", "PDF", and "Print". Below the search form is a table listing various medicines. The table has columns for Name, Price, Quantity, Update, and Remove. Each row contains a medicine name, its price, and quantity, followed by "Update" and "Remove" buttons. The medicines listed are Antibiotic, Antidepressant, Antipsychotic, Antiseptic, Aspirin, Benzodiazepines, Cytotoxics, Decongestants, Diuretics, and Expectorant. At the bottom of the table, there is a footer with links for First, Previous, Next, and Last.

Name	Price	Quantity	Update	Remove
Antibiotic	RM 123	10	<button>Update</button>	<button>Remove</button>
Antidepressant	RM 100	10	<button>Update</button>	<button>Remove</button>
Antipsychotic	RM 50	10	<button>Update</button>	<button>Remove</button>
Antiseptic	RM 20	15	<button>Update</button>	<button>Remove</button>
Aspirin	RM 50	10	<button>Update</button>	<button>Remove</button>
Benzodiazepines	RM 70	10	<button>Update</button>	<button>Remove</button>
Cytotoxics	RM 60	10	<button>Update</button>	<button>Remove</button>
Decongestants	RM 50	10	<button>Update</button>	<button>Remove</button>
Diuretics	RM 65	10	<button>Update</button>	<button>Remove</button>
Expectorant	RM 50	10	<button>Update</button>	<button>Remove</button>

*Figure 98: Search Medicine*

By clicking on the search button from the overview dashboard, it will automatically bring the staff to the search page when they can search for the name of the medicine, price, quantity as well as the option to update or remove a certain medicine. The staff can also search the amount of result to display on the screen.

## 16.4.6 Delete Medicine

The screenshot shows the Valley Clinic software interface. On the left is a sidebar with icons for Dashboard, Medicine (which is selected and highlighted in blue), Appointment, and Logout. The main area is titled "Medicine" and contains a table of medicine entries. The table has columns for Name, Price, Quantity, Update, and Remove. Each row represents a different medicine with its details. At the top of the table are buttons for Copy, CSV, Excel, PDF, and Print. There is also a "Search" input field and a "New Medicine" button. Below the table, it says "Showing 1 to 10 of 15 entries" and provides navigation links for First, Previous, Next, and Last.

Name	Price	Quantity	Update	Remove
Antibiotic	RM12	10	<button>Update</button>	<button>Remove</button>
Antidepressant	RM 100	10	<button>Update</button>	<button>Remove</button>
Antipsychotic	RM 50	10	<button>Update</button>	<button>Remove</button>
Antiseptic	RM 20	15	<button>Update</button>	<button>Remove</button>
Aspirin	RM 50	10	<button>Update</button>	<button>Remove</button>
Benzodiazepines	RM 70	10	<button>Update</button>	<button>Remove</button>
Cytotoxics	RM 60	10	<button>Update</button>	<button>Remove</button>
Decongestants	RM 50	10	<button>Update</button>	<button>Remove</button>
Diuretics	RM 65	10	<button>Update</button>	<button>Remove</button>
Expectorant	RM 50	10	<button>Update</button>	<button>Remove</button>

*Figure 99: Delete Medicine*

When the staff would like to delete any medicine, all the staff must do is just click on the medicine tab and delete any medicine that is not needed anymore by clicking on the remove button.

### 16.4.7 Check Appointment

The screenshot shows the 'Check Appointment' page of the Valley Clinic staff dashboard. At the top left is the Valley Clinic logo with a heart icon. On the right is a user profile icon labeled 'test'. The main area has a title 'Check Appointment' and a search bar. Below is a table with columns: AppID, Acc\_ID, DocID, App\_date, App\_slot, and App\_status. The table contains 10 entries from 1 to 10. At the bottom of the table are links for First, Previous, Next, and Last.

AppID	Acc_ID	DocID	App_date	App_slot	App_status
1	1	1	2021-01-26	1:30 PM	1
2	2	2	2021-02-22	4:16 AM	0
3	3	3	2021-03-24	10:47 PM	0
4	4	4	2021-04-07	6:43 AM	0
5	5	5	2021-05-04	2:19 AM	0
6	6	6	2021-06-25	7:18 PM	0
7	7	7	2021-07-11	8:20 PM	0
8	8	8	2021-08-04	12:31 AM	1
9	9	9	2021-09-26	7:18 PM	1
10	10	10	2021-10-23	4:44 AM	0

Figure 100: Check Appointment

The second function on the staff dashboard is appointment where the staff can check all the appointments for the day as well as the slots of check appointment, status, patients ID and doctor ID. However, the staffs are not able to edit any data of the appointment as the appointment are booked by the patient themselves.

## 16.4.8 Logout

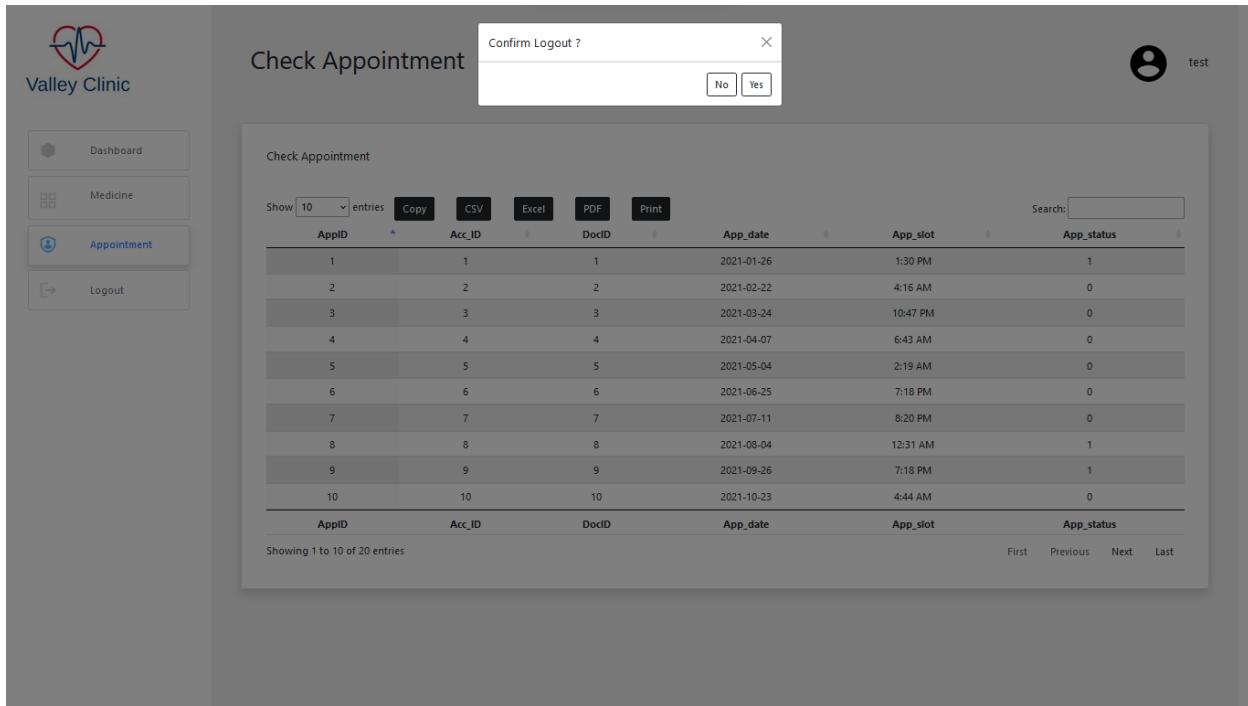


Figure 101: Staff Logout Page

Finally, when the staff would like to logout of the system then they would need to click on the logout button on the navigation panel which will bring out the confirm logout pop out. When the staff clicks on yes then the system will redirect the page back to the staff login page and when the staff clicks on not then the system will not redirect the page instead it will go back to the current session page.

# 17.0 Test Plan (Unit Testing)

## 17.1 Customer Module

Valley Clinic Management System	Valley Clinic Management System						
Component Name:	Customer Component						
Tester Name:	Hooi Kai Jun/Sam Zheng Yew						
Date of Creation:	3/2/2022						
Date of Review:	4/13/2022						
Test ID	Test Objectives	Precondition	Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
C_0001	Successful user login to the Valley Clinic Management System.	A valid user login.	In the login panel: 1. Enter the Username 2. Enter Password 3. Click Login Button	A valid Username and Password.	The user is logged successfully. The main page will be displayed on the screen.	User is logged in successfully. The main page will be displayed on the screen.	Pass
C_0002	Unsuccessful user login to the Valley Clinic Management System.	Invalid user login	In the login panel: 1. Enter the Username 2. Enter Password 3. Click Login Button	Invalid Username and Password.	A message box will appear stating that "Wrong username or Password has been entered"	A message will be displayed stating that "Wrong username or Password has been entered"	Fail
C_0003	Successfully register user to Valley Clinic Management System	Unsuccessful login/ User first time interacting with the system	In the sign-up panel: 1. Enter Username 2. Enter Password Repeat Password for confirmation Click Sign-up button	A valid Username and Password. 3. Example, Username: test 4. Password: test123	The user will successfully registered their account and will be redirected to the homepage.	The user successfully registered via the provided username and password. The user is also redirected to the homepage of the web system.	Pass
C_0004	Unsuccessful registration of user account to Valley Clinic Management System	Unsuccessful login/ User first time interacting with the system	In the sign-up panel: 1. Enter Username 2. Enter Password Repeat Password for confirmation Click Sign-up button	An invalid set of username and password. For example, an username or password combination where the username is already taken. Username: test Password: sam123	An error message will be displayed notifying the user that the provided username has already been taken.	The user is notified on the duplication of username from the provided username with an existing user in the system. Thus, an error message is displayed.	Pass
C_0005	User search treatment and services successfully	User navigate to treatment and services page with the name of the treatment or medical condition they wish to search	In the treatment and services page: 1. User enter the treatment or medical condition name into the search bar. 2. User click the magnifying glass icon or press "Enter" key to search keyword provided.	An existing record of medical condition of choice that is available in the database. For example, medical condition: Asthma, Anesthesiology, and etc.	A gallery which consists of the medical condition as search by the user and in some cases where similar keywords are matched with the records in the database, a gallery which consists of an image as well as the medical condition name will be displayed.	A gallery which consists of similar keywords as per user's search is shown with an image as a reference and the medical condition name associated with user's searched keyword.	Pass
C_0006	User search treatment and services unsuccessfully	User navigate to treatment and services page with the name of the treatment or medical condition they wish to search	In the treatment and services page: 1. User enter the treatment or medical condition name into the search bar. 2. User click the magnifying glass icon or press "Enter" key to search keyword provided.	An invalid name or keyword which is not recognized as stored in the database. For example, some unknown disease or medical condition that had not yet been inserted into the system. For example, measles.	An error message will be displayed notifying the user that the provided keyword does not match with information stored in the database.	An error message is displayed notifying the user on the unsuccessful search.	Pass
C_0007	User successfully contact staff administrator via contact us form	User has to be navigated to contact us page via the navigation bar by clicking on the contact us tab	In the contact us page: 1. User enter their full name. 2. User enter their contact number. 3. User enter their email address. 4. User enter their message. 5. User click Submit.	A valid set of input values such as the user's name, contact number, email address, and their text message.	Upon submission of form, the user will be prompted an alert asking for confirmation on form submission. If the user click "OK", the form will be submitted and the user will be redirected back to the current page. Whereas, if the user select "cancel" button, the user will be redirected back to the current page without submitting the form.	Once the user had keyed-in the respective input field, the user's form is submitted to Valley Clinic staff administration for review and the user is redirected back to the current page while users who select "cancel" is directly redirected back to the current page with the information still on the input field.	Pass

C_0008	User unsuccessful contact staff administrator using the contact us feature	User has to be navigated to contact us page via the navigation bar by clicking on the contact us tab	In the contact us page: 1. User enter their full name. 2. User enter their contact number. 3. User enter their email address. 4. User enter their message. 5. User click on submit button.	An invalid input value which consist of unanswered fields or invalid formats.	An error message will be prompt notifying the user on the incorrect formatting such as email address not having the symbol "@" or if there is any unanswered field.	Upon submission, user who left the fields unanswered are notified on the action needed to complete the form for submission.	Pass
C_0009	User successfully make a booking.	User are required to be in possession of an account registered in the system as well as being login into the system. The user are to navigate into the booking page as well through the navigation bar by selecting Online Booking dropdown list and into Bookings.	In the Booking page: 1. User enter their full name. 2. User enter their contact number. 3. User select the available doctor they wish to make appointment with. 3. User select an appointment date. 4. User click "Next". 5. User select any of the preferable booking slots given. 6. User click on submit. In the confirm booking page: 1. User will check the information previously entered.	A valid set of input which consist of user's full name, contact number, doctor available, appointment date, and chosen time slots. For example;  Full name: John Doe Contact Number: 0129239876 Doctor Available: Lawry Barracks Appointment date: 14/03/2022 Chosen time slot: 8.30	User will be prompt to the next page which is the confirmation page for their booking.	User is directed to the confirmation page and further action can be taken to make payment.	Pass
C_0009	User successfully make a booking.	User are required to be in possession of an account registered in the system as well as being login into the system. The user are to navigate into the booking page as well through the navigation bar by selecting Online Booking dropdown list and into Bookings.	Entered. 2. If no further changes are to be made, the user will select "confirm booking" button. In the Payment page: 1. if the user wish to make payment later, the user can click on pay later button. 2. If the user wish to pay now, the user will have to click on pay now. 3. User select payment method. 4. If credit card payment is chosen, user are required to enter their credit card information. 5. User click "Proceed to Checkout" button.]	A valid set of input which consist of user's full name, contact number, doctor available, appointment date, and chosen time slots. For example;  Full name: John Doe Contact Number: 0129239876 Doctor Available: Lawry Barracks Appointment date: 14/03/2022 Chosen time slot: 8.30	User will be prompt to the next page which is the confirmation page for their booking.	User is directed to the confirmation page and further action can be taken to make payment.	Pass
C_0010	User unsuccessful make a booking	User are required to be in possession of an account registered in the system as well as being login into the system. The user are to navigate into the booking page as well through the navigation bar by selecting Online Booking dropdown list and into Bookings	In the Booking page: 1. User enter their full name. 2. User enter their contact number. 3. User select the available doctor they wish to make appointment with. 3. User select an appointment date. 4. User click "Next". 5. User select any of the preferable booking slots given. 6. User click on submit.	An invalid set of input values such as incorrect data types or empty fields	The system will automatically prevent the user from submitting the form. An error message with the action needed to be performed to fix the issued will be displayed.	The user are not allowed to submit the form past the booking page and an error message is prompted notifying the user on the invalid entry as well as a set of action needed to be performed to correct the situation.	Pass

C_0011	User selecting an invalid time slot	User are required to be in possession of an account registered in the system as well as being login into the system. The user are to navigate into the booking page as well through the navigation bar by selecting Online Booking dropdown list and into Booking	In the Booking page: 1. User enter their full name. 2. User enter their contact number. 3. User select the available doctor they wish to make appointment with. 3. User select an appointment date. 4. User click "Next". 5. User select any of the preferable booking slots given. 6. User click on submit.	User select an already booked time slot.	The system will automatically prevent the user from selecting booked slots on the given date.	Time slots that are booked by other users are disabled and unable to be selected by the user.	Pass
C_0012	User selecting an invalid date for appointment	User are required to be in possession of an account registered in the system as well as being login into the system. The user are to navigate into the booking page as well through the navigation bar by selecting Online Booking dropdown list and into Booking	In the Booking page: 1. User enter their full name. 2. User enter their contact number. 3. User select the available doctor they wish to make appointment with. 3. User select an appointment date. 4. User click "Next". 5. User select any of the preferable booking slots given. 6. User click on submit.	User selecting the date of appointment from past dates.	The system will automatically prevent the user from selecting previous date.	Date of the past are not available for selecting and only the day of tomorrow can be selected.	Pass
C_0013	User unsuccessfully make payment via credit card	User are required to be in possession of an account registered in the system as well as being login into the system. The user are to navigate into the booking page as well through the navigation bar by selecting Online Booking dropdown list and into Booking.	In the Booking page: 1. User enter their full name. 2. User enter their contact number. 3. User select the available doctor they wish to make appointment with. 4. User select an appointment date. 5. User click "Next". 6. User select any of the preferable booking slots given. 7. User click on submit. In the confirm booking page: 1. User will check the information previously entered. 2. If no further changes are to be made..the	User provide an invalid input data type for credit card number, card name, expiry date, and CVV or any input field are left unanswered.	The system will prompt an error on the invalid data type or if the conditions are not met such as the card number are to be in a 16 digit figure, the CVV being in 3 figure and the expiry date are to be in the future date.	The system prompt the user with an error message notifying them on the invalid input as well as the correction method to be done such as the card number are to be in a 16 digit figure, the CVV being in 3 figure and the expiry date are to be in the future date.	Pass
C_0013	User unsuccessfully make payment via credit card	User are required to be in possession of an account registered in the system as well as being login into the system. The user are to navigate into the booking page as well through the navigation bar by selecting Online Booking dropdown list and into Booking.	entered. 2. If no further changes are to be made, the user will select "confirm booking" button. In the Payment page: 1. if the user wish to make payment later, the user can click on pay later button. 2. If the user wish to pay now, the user will have to click on pay now. 3. User select payment method. 4. If credit card payment is chosen, user are required to enter their credit card information. 5. User click "Proceed to Checkout" button.,	User provide an invalid input data type for credit card number, card name, expiry date, and CVV or any input field are left unanswered.	The system will prompt an error on the invalid data type or if the conditions are not met such as the card number are to be in a 16 digit figure, the CVV being in 3 figure and the expiry date are to be in the future date.	The system prompt the user with an error message notifying them on the invalid input as well as the correction method to be done such as the card number are to be in a 16 digit figure, the CVV being in 3 figure and the expiry date are to be in the future date.	Pass

# User Acceptance Testing

Name of Tester: Hooi Kai Jun

## Position: Programmer

Date: 10/3/2022

## 17.2 Administrator Module

Valley Clinic Management System		Valley Clinic Management System
Component Name:	Admin Component	
Tester Name:	Teo Kai Yii	
Date of Creation:	2/3/2022	
Date of Review:	13/4/2022	

Figure 86: Admin Component Test Plan

Test ID	Test Objectives	Precondition	Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
A_0001	Successful admin login to the clinic management system.	A valid admin login.	In the login panel: 1. Enter the Username 2. Enter Password 3. Click Login Button	A valid Username and Password.	The admin is logged in successfully. The admin dashboard will be displayed on the screen.	Admin provided the accurate username and password. Admin successfully logged in and was redirected to the admin dashboard	Pass
A_0002	Unsuccessful admin login to the clinic management system.	Invalid admin login	In the login panel: 1. Enter the Username 2. Enter Password 3. Click Login Button	Invalid Username and Password.	An alert message will appear stating that "Incorrect Username or Password"	Admin provided the invalid username and password. Admin unsuccessful login and was able to an alert message.	Pass
A_0003	Admin search treatment and services successfully.	Treatment and services search result shown.	In the left navigation panel: 1. Click Treatment & Services 2.Enter treatment name, specialist category or consultation fee range in search bar	Treatment and services that exist in database.	The admin is able to view the search result.	Admin provided the treatment and services that exist in database. The admin is able to search treatment and services successfully.	Pass
A_0004	Admin unsuccessful search treatment and services.	Treatment and services search result empty.	In the left navigation panel: 1. Click Treatment & Services 2.Enter treatment name, specialist category or consultation fee range in search bar	Treatment and services that did not exist in database.	A message will appear in the search result stating that "No matching records found"	Admin provided the treatment and services that did not exist in database. The admin notice with a message of "No matching records found" in the search result.	Pass
A_0005	Admin delete treatment and services successfully.	Treatment and services successfully removed.	In the left navigation panel: 1. Click Treatment & Services 2.Click the remove button on the respective row that admin want to delete	Treatment and services that exist in database.	An success message will appear "Treatment and Services Successfully Deleted!"	Treatment and services successfully deleted by using the treatment ID. A success message is provided to notice admin.	Pass
A_0006	Admin unsuccessful delete treatment and services.	Treatment and services are not removed.	In the left navigation panel: 1. Click Treatment & Services 2.Click the remove button on the respective row that admin want to delete	Invalid treatment and services exist in database.		Treatment and services are not deleted due to many treatment share the same treatment ID. An alert message is provided to notice admin.	Pass
A_0007	Admin successfully create new treatment and services.	Treatment and services successfully created.	In the Search Treatment and services Page: 1.Click "New Treatment and services" 2. Fill up Title, Description, Specialist Category, Consultation Fee Range and Upload picture	Valid Title, Description, Specialist Category, Consultation Fee Range and Upload valid picture format	An success message will appear "Treatment and services Successfully Created!"	Admin provided valid Title, Description, Specialist Category, Consultation Fee Range and Upload valid picture format. System validate informations. A success message is provided to notice admin	Pass
A_0008	Admin unsuccessful create new treatment and services.	Treatment and services are not created.	In the Search Treatment and services Page: 1.Click "New Treatment and services" 2. Fill up Title, Description, Specialist Category, Consultation Fee Range and Upload picture	Invalid Title, Description, Specialist Category, Consultation Fee Range and Upload invalid picture format	An alert message will appear "Sorry, there was an error uploading your file. Please check again"	Admin provided invalid Title, Description, Specialist Category, Consultation Fee Range and Upload invalid picture format. System validate informations. An alert message is provided to notice admin.	Pass
A_0009	Admin update treatment and services successfully.	Treatment and services successfully updated.	In the left navigation panel: 1. Click Treatment & Services 2.Click the update button on the respective row that admin want to update	Valid Title, Description, Specialist Category, and Consultation Fee Range	An success message will appear "Treatment and services Successfully Updated!"	Admin provided valid Title, Description, Specialist Category, and Consultation Fee Range. System validate informations. A success message is provided to notice admin	Pass
A_0010	Admin unsuccessful update treatment and services.	Treatment and services are not updated.	In the left navigation panel: 1. Click Treatment & Services 2.Click the update button on the respective row that admin want to update	Invalid Title, Description, Specialist Category, or Consultation Fee Range	An alert message will appear "Please Try Again!"	Admin provided invalid Title, Description, Specialist Category, or Consultation Fee Range. System validate informations. An alert message is provided to notice admin.	Pass
A_0011	Admin search admin account successfully.	Admin accounts search result shown.	In the left navigation panel: 1. Click Admin	Admin accounts that exist in database.	The admin is able to view the search result.	Admin provided the admin username that exist in database. The admin is able to search admin accounts successfully.	Pass
A_0012	Admin unsuccessful search admin account.	Admin accounts search result empty.	In the left navigation panel: 1. Click Admin	Admin accounts that did not exist in database.	A message will appear in the search result stating that "No matching records found"	Admin provided the admin username that did not exist in database. The admin notice with a message of "No matching records found" in the search result.	Pass
A_0013	Admin delete admin account successfully.	Admin account successfully removed.	In the left navigation panel: 1. Click Admin 2.Click the remove button on the respective row that admin want to delete	Admin account that exist in database.	An success message will appear "Admin account Successfully Deleted!"	Admin account successfully deleted by using the admin ID. A success message is provided to notice admin.	Pass
A_0014	Admin unsuccessful delete admin account.	Admin account are not removed.	In the left navigation panel: 1. Click Admin 2.Click the remove button on the respective row that admin want to delete	invalid admin account exist in database.	An alert message will appear "Please Try Again!"	Admin account are not deleted due to there are more than one result while searching with the admin ID. An alert message is provided to notice admin.	Pass
A_0015	Admin create new admin account successfully.	Admin account successfully created.	In the Search Admin Page: 1.Click "New Account" 2. Fill up username, password, and confirm password	Unique username, valid password, confirm password that matches password.	An success message will appear "Admin account Successfully Created!"	Admin provided unique username, valid password, confirm password that matches password. System validate informations. A success message is provided to notice admin	Pass

Figure 87: Admin Component Test Plan

A_0016	Admin unsuccessful create new admin account due to repeated username.	Admin account are not created.	In the Search Admin Page: 1.Click "New Account" 2. Fill up username, password, and confirm password	Repeated username, valid password, confirm password that matches password	An alert message will appear "Username already taken! Please choose another username."	Admin provided repeated username, valid password, confirm password that matches password. System validate informations. An alert message is provided to notice admin.	Pass
A_0017	Admin unsuccessful create new admin account due to password and confirm password did not match.	Admin account are not created.	In the Search Admin Page: 1.Click "New Account" 2. Fill up username, password, and confirm password	Unique username, valid password, confirm password that do not match password	An alert message will appear "Please ensure your password and confirm password are matched."	Admin provided unique username, valid password, confirm password that do not match password. System validate informations. An alert message is provided to notice admin.	Pass
A_0018	Admin search staff account successfully.	Staff accounts search result shown.	In the left navigation panel: 1. Click Staff 2. Enter staff username in search bar	Staff accounts that exist in database.	The admin is able to view the search result.	Admin provided the staff username that exist in database. The admin is able to search staff accounts successfully.	Pass
A_0019	Admin unsuccessful search staff account.	Admin accounts search result empty.	In the left navigation panel: 1. Click Staff 2. Enter staff username in search bar	Staff accounts that did not exist in database.	A message will appear in the search result stating that "No matching records found"	Admin provided the staff username that did not exist in database. The admin notice with a message of "No matching records found" in the search result.	Pass
A_0020	Admin delete staff account successfully.	Staff account successfully removed.	In the left navigation panel: 1. Click Staff 2. Click the remove button on the respective row that admin want to delete	Staff account that exist in database.	An success message will appear "Staff account Successfully Deleted !"	Staff account successfully deleted by using the staff ID. A success message is provided to notice admin.	Pass
A_0021	Admin unsuccessful delete staff account.	Staff account are not removed.	In the left navigation panel: 1. Click Staff 2. Click the remove button on the respective row that admin want to delete	Invalid Staff account exist in database.	An alert message will appear "Please Try Again!"	Staff account are not deleted due to there are more than one result while searching with the staff ID. An alert message is provided to notice admin.	Pass
A_0022	Admin create new staff account successfully.	Staff account successfully created.	In the Search Staff Page: 1.Click "New Account" 2. Fill up username, password, and confirm password	Unique username, valid password, confirm password that matches password	An success message will appear "Admin account Successfully Created !"	Admin provided unique username, valid password, confirm password that matches password. System validate informations. A success message is provided to notice admin	Pass
A_0023	Admin unsuccessful create new staff account due to repeated username.	Staff account are not created.	In the Search Staff Page: 1.Click "New Account" 2. Fill up username, password, and confirm password	Repeated username, valid password, confirm password that matches password	An alert message will appear "Username already taken! Please choose another username."	Admin provided repeated username, valid password, confirm password that matches password. System validate informations. An alert message is provided to notice admin.	Pass
A_0024	Admin unsuccessful create new staff account due to password and confirm password did not match.	Staff account are not created.	In the Search Staff Page: 1.Click "New Account" 2. Fill up username, password, and confirm password	Unique username, valid password, confirm password that do not match password	An alert message will appear "Please ensure your password and confirm password are matched."	Admin provided unique username, valid password, confirm password that do not match password. System validate informations. An alert message is provided to notice admin.	Pass
A_0025	Admin search doctor account successfully.	Doctor accounts search result shown.	In the left navigation panel: 1. Click Doctors 2. Enter doctor name, specialist category or contact number in search bar	Doctor accounts that exist in database.	The admin is able to view the search result.	Admin provided the doctor information that exist in database. The admin is able to search treatment and services successfully.	Pass
A_0026	Admin unsuccessful search doctor account.	Doctor accounts search result empty.	In the left navigation panel: 1. Click Doctors 2. Enter doctor name, specialist category or contact number in search bar	Doctor accounts that did not exist in database.	A message will appear in the search result stating that "No matching records found"	Admin provided the doctor information that did not exist in database. The admin notice with a message of "No matching records found" in the search result.	Pass
A_0027	Admin delete doctor account successfully.	Doctor account successfully removed.	In the left navigation panel: 1. Click Doctor 2. Click the remove button on the respective row that admin want to delete	Doctor account that exist in database.	An success message will appear "Doctor account Successfully Deleted !"	Doctor account successfully deleted by using the staff ID. A success message is provided to notice admin.	Pass
A_0028	Admin unsuccessful delete doctor account.	Doctor account are not removed.	In the left navigation panel: 1. Click Doctor 2. Click the remove button on the respective row that admin want to delete	Invalid Doctor account exist in database.	An alert message will appear "Please Try Again!"	Doctor account are not deleted due to there are more than one result while searching with the staff ID. An alert message is provided to notice admin.	Pass
A_0029	Admin create new doctor account successfully.	Doctor account successfully created.	In the Search Doctor Page: 1.Click "New Account" 2. Fill up username, password, and confirm password	Unique username, valid password, confirm password that matches password	An success message will appear "Admin account Successfully Created !"	Admin provided unique username, valid password, confirm password that matches password. System validate informations. A success message is provided to notice admin	Pass
A_0030	Admin unsuccessful create new staff account due to repeated username.	Doctor account are not created.	In the Search Doctor Page: 1.Click "New Account" 2. Fill up username, password, and confirm password	Repeated username, valid password, confirm password that matches password	An alert message will appear "Username already taken! Please choose another username."	Admin provided repeated username, valid password, confirm password that matches password. System validate informations. An alert message is provided to notice admin.	Pass
A_0031	Admin unsuccessful create new staff account due to password and confirm password did not match.	Doctor account are not created.	In the Search Doctor Page: 1.Click "New Account" 2. Fill up username, password, and confirm password	Unique username, valid password, confirm password that do not match password	An alert message will appear "Please ensure your password and confirm password are matched."	Admin provided unique username, valid password, confirm password that do not match password. System validate informations. An alert message is provided to notice admin.	Pass
A_0032	Admin successfully logout admin dashboard.	Admin wants to logout.	In the left navigation panel: 1. Click Logout In the pop out message box: 2. Click Yes	Yes or no response by admin.	Admin successfully logout the admin dashboard.	Admin click the logout button and confirm logout by choosing the yes option. Admin was redirected to the logout success page and see a message of "Logged Out Successfully".	Pass
A_0033	Admin unsuccessful logout admin dashboard.	Admin accidentally clicked the logout button.	In the left navigation panel: 1. Click Logout the pop out message box: 2. Click No	Yes or no response by admin.	Admin remains in the current page.	Admin click the logout button and confirm logout by choosing the no option. Admin was not redirected to the logout success page.	Pass

Figure 88: Admin Component Test Plan

## User Acceptance Testing

Name of Tester: Teo Kai Yii

Position: Project Manager

Date: 10/3/2021

### Interface

TC NO	Criteria	Excellent	Good	Average	Poor
1	Are the website's interfaces attractive and simple to use?	/			
2	Are the colors used in the screen making it more user-friendly?		/		
3	Are the columns and other elements of the same size aligned properly?		/		
4	Is the data structured in a logical manner?			/	
5	Are the elements on the screen labeled with their respective functions?		/		

### Consistency

TC NO	Criteria	Excellent	Good	Average	Poor
1	Are symbols, icons, and other graphical representations applied consistently?	/			
2	Is the format of the user input consistent?	/			
3	Are all the colors used on the website the consistent?	/			
4	Is the data displayed in the application accurate?	/			

## Performance

TC NO	Criteria	Yes	No	Remark
1	Is the alert working properly and without any errors or delays?	/		
2	Is the website crashing or closing unexpectedly while in use?		/	
3	Is your PC battery draining quicker than normal while you're running this website?		/	
4	Is it time-consuming to verify your account and password?		/	
5	Do you think it takes a long time for the web server to respond?		/	
6	Do you believe your data was stored in the shortest possible time?		/	

## Information Feedback

TC NO	Criteria	Yes	No	Remark
1	Are all directions in each section properly stated?	/		
2	Is the user prompted with an acknowledgment message when a specific function is completed?	/		
3	Are the error message stating what the problem is?		/	Some parts just write “Please Try again”
4	Is there a prompt for confirmation of the modifications made by the user?	/		
5	Is the user informed about the time it takes to complete a procedure that causes a delay by the system?		/	

## Error Prevention and Correction

TC NO	Criteria	Yes	No	Remark
1	When the system identifies a problem, does it display an error message?	/		Sometimes error may show
2	Does the modal appear when you click the logout button?	/		
3	Is it possible for an administrator to remove a user account?	/		Admin can only remove user account whenever requested by the clinic management.
4	Is the displayed error message simple to understand?	/		
5	Does the error message explain what the error is?	/		Some error message is not clear.
6	Is there a message prompt with an option prompt?	/		
7	Is the user's instruction to solve the issue clear?	/		
8	Is the website, in general, free of errors or breakdowns?	/		

## User Usability

TC NO	Criteria	Yes	No	Remark
1	Is the data input process time-consuming?	/		Filling up forms like treatment and services form may take some time.
2	Is there any limitation imposed by the uncertainty of where and how the user's information should be entered?		/	
3	Is there an issue with the website response speed, such as it takes a long time for you to figure out what's going on?		/	
4	Is the website equipped with a Help feature?		/	Users could reach out to the clinic if they faced any issue.
5	Is there a function that is as difficult to use and has unclear objectives?		/	
6	Do you need further guidance or an instructor to show how to use the system?	/		Installation guide is required to access the website.
7	Is this system simple to use for you?	/		

## Security

TC NO	Criteria	Excellent	Good	Average	Poor
1	Do you think the login process is safe?	/			
2	Do you think your personal information is safe?		/		
3	Do you think it's safe to book an appointment?		/		
4	Do you believe that using this website is safe and secure?		/		

Comment From Tester	Please fix all the bugs we have now. Improve the website interactivity with user to achieve a user-friendly system.
Action taken by the developer	More buttons and clearer instructions have been implemented into the system to provide better flow.

### 17.3 Doctor Module

<b>Project Name:</b>	Valley Clinic Management System						
<b>Component Name:</b>	Doctor Component						
<b>Tester Name:</b>	Tan Kar Wei						
<b>Date of Creation:</b>	3/3/2022						
<b>Date of Review:</b>	4/10/2022						
Test ID	Test Objectives	Precondition	Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
T_0001	Successful login to the system	A registered doctor login	At the login page: 1) Enter username 2) Enter password 3) Click on login button	A valid registered doctor account	The system will notify the user login successfully and direct the user to the system homepage	A alert box appear notifying the user "Login Successfully" and direct the user to the system homepage	Pass
T_0002	Unsuccessful login to the system	A non-registered doctor login	At the login page: 1) Enter username 2) Enter password 3) Click on login button	An unregistered doctor account	The system will deny the entry of the user to the system and notify the user that either username or password entry is incorrect. Lastly, the system will direct the user back to the system login page	A alert box appear notifying the user "Incorrect username or password" and direct the user back to the login page	Pass
T_0003	Doctor search appointment successfully	Valid data of the table	In the side navigation bar: 1) Click on 'Appointment' 2) Enter search value on the search bar above the table	Data that exist in the database	The system will show the most relevant result on the table	The system will show the most relevant result on the table	Pass
T_0004	Doctor search appointment unsuccessfully	Valid data of the table	In the side navigation bar: 1) Click on 'Appointment' 2) Enter search value on the search bar above the table	Data that does not exist in the database	The system will not show any data and notify the users that no records were found	The system will not show any data and appear a line of "No matching records found"	Pass
T_0005	Doctor search patient account successfully	Valid data of the table	In the side navigation bar: 1) Click on 'Treatment Record' 2) Enter search value on the search bar above the table	Data that exist in the database	The system will show the most relevant result on the table	The system will show the most relevant result on the table	Pass
T_0006	Doctor search patient account unsuccessfully	Valid data of the table	In the side navigation bar: 1) Click on 'Treatment Record' 2) Enter search value on the search bar above the table	Data that does not exist in the database	The system will not show any data and notify the users that no records were found	The system will not show any data and appear a line of "No matching records found"	Pass

Figure 89 Doctor Component Test Plan

T_0007	Doctor search patient appointment unsuccessfully	Valid data of the table	In the side navigation bar: 1) Click on 'Treatment Record' 2) Click on the 'View' button beside the patient 3) Enter search value on the search bar above the table	Data that exist in the database	The system will show the most relevant result on the table	The system will show the most relevant result on the table	Pass
T_0008	Doctor search patient appointment unsuccessfully	Valid data of the table	In the side navigation bar: 1) Click on 'Treatment Record' 2) Click on the 'View' button beside the patient 3) Enter search value on the search bar above the table	Data that does not exist in the database	The system will not show any data and notify the users that no records were found	The system will not show any data and appear a line of "No matching records found"	Pass
T_0009	Doctor update treatment record successsfully	Exist appointment info	In the treatment record page 2: 1) Click on 'Edit' button 2) Enter text value on the input box 3) Click on 'Update' button	Text value	The system will notify the user updated successfully and close the edit modal	A alert box appear notifying the user "Record updated successfully" and close the modal box	Pass
T_0010	Doctor update treatment record unsuccesssfully	Exist appointment info	In the treatment record page 2: 1) Click on 'Edit' button 2) Enter text value on the input box 3) Click on 'Update' button	Empty value	The system will deny the update of the record and notify the user to fill in some value.	A alert box appear notifying the user "Notes is required"	Pass
T_0011	Doctor edit profile details successsfully	successfully connected to database	At the account page: 1)Click on the 'Edit Profile' button 2) Fill value into the input box 3) Click on "Submit" button	All input box are filled	The system will notify the user update successfully and direct the user back to the account page	A alert box appear notifying the user "Account has been updated" and direct the user back to the account page	Pass
T_0012	Doctor edit profile details unsuccesssfully	successfully connected to database	At the account page: 1)Click on the 'Edit Profile' button 2) Fill value into the input box 3) Click on "Submit" button	Input box left empty	The system will notify the user to fill up all the input box	Pop up with "Please fill out this filed" will appear on the empty input box	Pass
T_0013	Doctor edit profile details unsuccesssfully	unsuccessfully connect to database	At the account page: 1)Click on the 'Edit Profile' button 2) Fill value into the input box 3) Click on "Submit" button	All input box are filled	The system will notify the user update failed	A alert box appear notifying the user "Update failed"	Pass

Figure 90 Doctor Component Test Plan

T_0014	Doctor edit profile picture successfully	successfully connected to database	At the account page: 1)Move mouse to profile picture 2) Click on "Change Picture" 3) Upload a image 4)Click on "Upload"	Image uploaded to entry box	The system will notify the user update successfully and direct the user back to the account page	A alert box appear notifying the user "Profile Picture had been update" and direct the user back to the account page	Pass
T_0015	Doctor edit profile picture unsuccessfully	successfully connected to database	At the account page: 1)Move mouse to profile picture 2) Click on "Change Picture" 3) Upload a image 4)Click on "Upload"	Empty input box	The system will notify the user to fill up all the input box	Pop up with "Please select a file" will appear on the empty input box	Pass
T_0016	Doctor edit profile picture unsuccessfully	unsuccessfully connect to database	At the account page: 1)Move mouse to profile picture 2) Click on "Change Picture" 3) Upload a image 4)Click on "Upload"	Image uploaded to entry box	The system will notify the user update failed	A alert box appear notifying the user "Update failed"	Pass
T_0017	Doctor successfully logout from the system	Doctor wanted to logout	In the left navigation bar 1)Click "Logout" 2) Click "Logout" on the modal	Confirm response from doctor	The system dicrect the user back to the login page	The system dicrect the user back to the login page	Pass
T_0018	Doctor unsuccessful logout from the system	Doctor accidentally clicked on the logout button	In the left navigation bar 1)Click "Logout" 2) Click "Cancel" on the modal	Cancel response from doctor	The system will remain in the current page	The system will remain in the current page	Pass

Figure 91 Doctor Component Test Plan

## User Acceptance Testing

Name of Tester: Tan Kar Wei

Position: -

Date: 11/3/2022

### Interface

TC NO	Criteria	Excellent	Good	Average	Poor
1	Are the website's interfaces attractive and simple to use?		✓		
2	Are the colors used in the screen making it more user-friendly?	✓			
3	Are the columns and other elements of the same size aligned properly?		✓		
4	Is the data structured in a logical manner?		✓		
5	Are the elements on the screen labeled with their respective functions?	✓			

### Consistency

TC NO	Criteria	Excellent	Good	Average	Poor
1	Are symbols, icons, and other graphical representations applied consistently?		✓		
2	Is the format of the user input consistent?		✓		
3	Are all the colors used on the website the consistent?	✓			
4	Is the data displayed in the application accurate?	✓			

## Performance

TC NO	Criteria	Yes	No	Remark
1	Is the alert working properly and without any errors or delays?		✓	
2	Is the website crashing or closing unexpectedly while in use?		✓	
3	Is your PC battery draining quicker than normal while you're running this website?		✓	
4	Is it time-consuming to verify your account and password?		✓	
5	Do you think it takes a long time for the web server to respond?		✓	
6	Do you believe your data was stored in the shortest possible time?	✓		

## Information Feedback

TC NO	Criteria	Yes	No	Remark
1	Are all directions in each section properly stated?		✓	
2	Is the user prompted with an acknowledgment message when a specific function is completed?	✓		
3	Are the error message stating what the problem is?		✓	
4	Is there a prompt for confirmation of the modifications made by the user?	✓		
5	Is the user informed about the time it takes to complete a procedure that causes a delay by the system?		✓	

## Error Prevention and Correction

TC NO	Criteria	Yes	No	Remark
1	When the system identifies a problem, does it display an error message?		✓	
2	Does the modal appear when you click the logout button?	✓		
3	Is it possible for an administrator to remove a user account?	✓		
4	Is the displayed error message simple to understand?	✓		
5	Does the error message explain what the error is?		✓	
6	Is there a message prompt with an option prompt?		✓	
7	Is the user's instruction to solve the issue clear?		✓	
8	Is the website, in general, free of errors or breakdowns?	✓		

## User Usability

TC NO	Criteria	Yes	No	Remark
1	Is the data input process time-consuming?		✓	
2	Is there any limitation imposed by the uncertainty of where and how the user's information should be entered?	✓		
3	Is there an issue with the website response speed, such as it takes a long time for you to figure out what's going on?		✓	
4	Is the website equipped with a Help feature?		✓	

5	Is there a function that is as difficult to use and has unclear objectives?		✓	
6	Do you need further guidance or an instructor to show how to use the system?	✓		
7	Is this system simple to use for you?	✓		

## Security

TC NO	Criteria	Excellent	Good	Average	Poor
1	Do you think the login process is safe?	✓			
2	Do you think your personal information is safe?		✓		
3	Do you think it's safe to book an appointment?		✓		
4	Do you believe that using this website is safe and secure?		✓		

Comment From Tester	System may be <b>too</b> simple, does not have any fancy or advanced function.
Action taken by the developer	Make it more fancy by adding the side navigation bar

## 17.4 Staff Module

<b>Project Name:</b>	Valley Clinic Management System
<b>Component Name:</b>	Staff Component
<b>Tester Name:</b>	Nisharini Ganesan
<b>Date of Creation:</b>	2/3/2022
<b>Date of Review:</b>	13/4/2022

*Figure 120: Staff Component Test Plan*

Test ID	Test Objectives	Precondition	Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
T_0001	Successful staff login to the clinic management system.	Valid staff username and password	1. Staff enters username and password to enter into the system. 2. Login into the system is successful.	Valid username and password	Staff successfully enter username and password. The staff dashboard overview is shown.	Login successful	Pass
T_0002	Unsuccessful staff login to the clinic management system.	Invalid staff username and password	1. Staff enters username and password to enter into the system. 2. Login into the system is unsuccessful.	Invalid username and password	Staff unsuccessfully enter username and password. The main page of system is not shown instead it shows "Incorrect username or password.".	Login unsuccessful	Pass
T_0003	Staff successfully open medicine tab.	Medicine tab results are shown.	1. In the staff navigation panel, click on the medicine tab. 2. Details on the medicine tab will be displayed.	Medicine database exists.	The results from medicine table is shown.	Successful results	Pass
T_0004	Staff unsuccessfully open medicine tab.	Medicine tab results are not shown.	1. In the staff navigation panel, click on the medicine tab. 2. Details on the medicine tab will be displayed.	Medicine database does exists.	The results from medicine table is not shown.	Unsuccessful results	Pass
T_0005	Staff successfully add medicine.	Add medicine results are shown.	1. Under medicine tab, click on add medicine. 2. Add new medicine by filling up information.	Add medicine table exists in the database.	Page to insert new medicine is shown.	Add sucessful	Pass
T_0006	Staff unsuccessfully add medicine.	Add medicine results are not shown.	1. Under medicine tab, click on add medicine. 2. Add new medicine by filling up information.	Add medicine table does exists in the database.	Page to insert new medicine is not shown.	Add unsucessful	Pass
T_0007	Staff successfully edit medicine.	Edit medicine tab results are shown.	1. Under medicine tab, click on edit medicine. 2. Edit medicine information by changing medicine details.	Edit medicine table exist in the database.	Page to edit medicine information is shown.	Edit sucessful	Pass
T_0008	Staff unsuccessfully edit medicine.	Edit medicine tab results are not shown.	1. Under medicine tab, click on edit medicine. 2. Edit medicine information by changing medicine details.	Edit medicine table does exist in the database.	Page to edit medicine information is not shown.	Edit unsucessful	Pass
T_0009	Staff successfully search medicine.	Search medicine tab results are shown.	1. Under medicine tab, click on search medicine. 2. Search medicine by entering required information.	Search medicine table exist in the database.	Page to search medicine is shown.	Search sucessful	Pass

*Figure 121: Staff Component Test Plan*

T_0010	Staff unsuccessfully search medicine.	Search medicine tab results are not shown.	1. Under medicine tab, click on search medicine. 2. Search medicine by entering required information.	Search medicine table does exist in the database.	Page to search medicine is not shown.	Search unsuccesful	Pass
T_0011	Staff successfully delete medicine.	Delete medicine tab results are shown.	1. Under medicine tab, click on delete medicine. 2. Delete medicine by clicking on the said medicine.	Delete medicine table exist in database.	Page to delete medicine is shown.	Delete sucessful	Pass
T_0012	Staff unsuccessfully delete medicine.	Delete medicine tab results are not shown.	1. Under medicine tab, click on delete medicine. 2. Delete medicine by clicking on the said medicine.	Delete medicine table does exist in database.	Page to delete medicine is not shown.	Delete unsuccesful	Pass
T_0013	Staff successfully open appointment tab.	Appointment tab results are shown.	1. In the staff navigation panel, click on the appointment tab. 2. Details on the appointment tab will be displayed.	Appointment database exists.	The results from appointment table is shown.	Successful results	Pass
T_0014	Staff unsuccessfully open appointment tab.	Appointment tab results are not shown.	1. In the staff navigation panel, click on the appointment tab. 2. Details on the appointment tab will be displayed.	Appointment database does not exists.	The results from appointment table is not shown.	Unsuccessful results	Pass
T_0015	Staff successfully check appointment.	Check appointment tab results are shown.	1. Under appointment tab, click on check appointment. 2. Check all appointments for the said day.	Check appointment details are in the database.	Page to check appointment is shown.	View successful	Pass
T_0016	Staff unsuccessfully check appointment.	Check appointment tab results are not shown.	1. Under appointment tab, click on check appointment. 2. Check all appointments for the said day.	Check appointment details are not in the database.	Page to check appointment is not shown.	View unsuccessful	Pass
T_0017	Successful staff logout from clinic management system.	Staff would like to logout from the management system.	1. In the staff navigation panel, click on the logout tab. 2. Popup will display click yes.	Yes or no response from staff.	Staff successfully logout from the system.	Logout successful	Pass
T_0018	Unsuccessful staff logout from clinic management system.	Staff would not like to logout from the management system.	1. In the staff navigation panel, click on the logout tab. 2. Popup will display click no.	Yes or no response from staff.	Staff remains in the system.	Logout unsuccessful	Pass

*Figure 122: Staff Component Test Plan*

## User Acceptance Testing

Name of Tester: Nisharini Ganesan

Position: System Tester

Date: 11/3/2022

### Interface

TC NO	Criteria	Excellent	Good	Average	Poor
1	Is the website interface simple yet attractive?	/			
2	Are the colors used to design the website are user-friendly?		/		
3	Are the properties of the website aligning the same?		/		
4	Is the data structured in a logical manner?			/	
5	Are the functions on website labeled with its respectively element?		/		

### Consistency

TC NO	Criteria	Excellent	Good	Average	Poor
1	Are the symbols, icons, and other graphical representations used on the website applied consistently?		/		
2	Is the format of the user input consistent?		/		
3	Are all the colors used to make the website consistent?		/		
4	Is the data displayed in the website accurate?		/		

## Performance

TC NO	Criteria	Yes	No	Remark
1	Is the alert working properly and without any errors or delays?	/		
2	Is the website crashing or closing unexpectedly while in use?		/	
3	Is your computer performance slowing down quicker than normal while you're running this website?		/	
4	Is it time-consuming to verify your account username and password?		/	
5	Does it take a long time for the web server to respond?		/	
6	Are you sure that your data was stored in the shortest possible time?		/	

## Information Feedback

TC NO	Criteria	Yes	No	Remark
1	Are all directions in each section properly stated?	/		
2	Is the user prompted with an acknowledgment message when a specific function is completed?	/		
3	Are the error messages stating what the problem is with each section?		/	Incorrect input displayed.
4	Is there any prompt for confirmation of the modifications done by the user?	/		
5	Is the user informed about the time it takes to complete a procedure that causes a delay by the system?		/	

## Error Prevention and Correction

TC NO	Criteria	Yes	No	Remark
1	When the system identifies a problem, does it display an error message?	/		Not all error is being shown.
2	Is it possible for a staff to add new medicine?	/		
3	Is it possible for a staff to remove appointment?	/		Staff does not have the access to remove any appointment.
4	Is the displayed error message simple to understand?	/		
5	Does the error message explain what the error is?	/		
6	Is there any message prompt with an option prompt?	/		
7	Is the user's instruction to solve the issue clear to understand?	/		
8	Is the website, in general, free of errors or breakdowns?	/		

## User Usability

TC NO	Criteria	Yes	No	Remark
1	Is the data input process time-consuming?	/		Inputting data one at a time is time consuming
2	Is there any limitation imposed by the uncertainty of where and how the user's information should be entered?	/		
3	Is there an issue with the website response speed, such as it takes a	/		

	long time for you to figure out what's going on?			
4	Is the website equipped with a Help feature?	/		
5	Is there any function that is as difficult to use and has unclear objectives?	/		
6	Do you need further guidance on how to use the system?	/		
7	Is this system simple to use?	/		

## Security

TC NO	Criteria	Excellent	Good	Average	Poor
1	Do you think the login process is safe?	/			
2	Do you think it is safe to add new medicine for the clinic?	/			
3	Do you think it's safe to delete medicine from the clinic?	/			
4	Do you believe that using this website on a web server is safe and secure?	/			

Comment From Tester	Improve on the weakness of the website by looking at the results from the test above.
Action taken by the developer	More tabs and buttons created for a better and easier flow structure of the website.

## 18.0 Significant Source Code

### 18.1 Bootstrap

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
```

Figure 92: Bootstrap CSS

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>
```

Figure 93: Bootstrap JavaScript (Bundle)

Admin dashboard of Valley Clinic Management System is developed by using Bootstrap. Bootstrap is the world's most popular framework for developing flexible, mobile-first webpages with jsDelivr (Bootstrap, n.d.). jsDelivr is a free content delivery network (CDN) to load CSS and JavaScript remotely from the servers. In admin dashboard, a stylesheet link is placed at the <head> section to load CSS. As for the JavaScript, bundle version of JavaScript which include functions for some of the components. This JavaScript is placed right before the </body> tag to load JavaScript (Bootstrap, n.d.).

Additionally, Valley Clinic Management System also utilizes Bootstrap Compiled version instead of the CDN version. Upon downloading Bootstrap compiled zip file, the folders of which included in the Bootstrap folder can be simply drop into the project folder and similar to the CDN version, to connect the project with Bootstrap, just include the link of the stylesheet and jsDelivr into the HTML file.

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

Figure 94 Bootstrap CSS via Compiled download

```
<script src="../Bootstrap/js/bootstrap.bundle.js"></script>
```

Figure 95 Include Jsbundle from Compiled Bootstrap

## 18.2 Bootstrap Icon

```
<link rel="stylesheet" href=
"https://cdn.jsdelivr.net/npm/bootstrap-icons@1.8.1/font/bootstrap-icons.css"
>
```

*Figure 96: Bootstrap icon CDN*

This CDN is placed on the header of admin dashboard pages to access icons. Bootstrap icon consists over 1,600 icons which is free, high quality and open-source icon collection (Bootstrap Icons, n.d.).

### 18.3 Font Awesome Icon

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

*Figure 97 Font-Awesome Icons*

Font Awesome is a font and icon toolkit which is based on CSS and allows for ease of styling. Font Awesome is always free and is also an open source which provide various support in the community by subscribing to Font Awesome Pro. The link to Font Awesome toolkit is located at the head section of various HTML code and PHP files where the toolkit is linked to the particular page. Within the webpages, various locations can be seen that uses Font-Awesome's icons such as shown below in Figure 97. Furthermore, Font Awesome are also used as visual aids which provide graphical information alongside text, making the whole design cleaner and attractive.

```
<div class="service-container">
    <h2>Service Highlights</h2>
    <div class="highlight-container">
        <div class="highlight-gallery">
            <i class="fa fa-user-md" aria-hidden="true"></i>
            <p>Experienced doctors and nurses</p>
        </div>
        <div class="highlight-gallery">
            <i class="fa fa-hospital-o" aria-hidden="true"></i>
            <p>Hygenic and comfortable environment</p>
        </div>
        <div class="highlight-gallery">
            <i class="fa fa-stethoscope" aria-hidden="true"></i>
            <p>Latest medical equipment</p>
        </div>
        <div class="highlight-gallery">
            <i class="fa fa-address-book" aria-hidden="true"></i>
            <p>Booking and appointment services</p>
        </div>
    </div>
</div>
```

Figure 98 Example of Font-Awesome toolkit used in index.php

## 18.4 Chart.js

```
<script src=
"https://cdnjs.cloudflare.com/ajax/libs/Chart.j
s/3.7.1/chart.min.js"
></script>
```

Figure 99: *Chart.js CDN*

Chart.js allows JavaScript charting for designers and developers that is simple and adaptable. Chart.js helps to visualize data in 8 different methods of animation. To begin, a canvas is needed on our website. For responsiveness, it's best to provide the chart its own container (Chart.js, 2022). The JavaScript CDN of Chart.js is included in the HTML code right before the body tag.

## 18.5 DataTables

```
<!-- datatables -->
<link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/1.11.4/css/jquery.dataTables.min.css">
<script src="https://code.jquery.com/jquery-3.5.1.js"></script>
<script src="https://cdn.datatables.net/1.11.4/js/jquery.dataTables.min.js"></script>
<!-- additional -->
<link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/buttons/2.2.2/css/buttons.dataTables.min.css">
<script src="https://cdn.datatables.net/buttons/2.2.2/js/dataTables.buttons.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/pdfmake.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/vfs_fonts.js"></script>
<script src="https://cdn.datatables.net/buttons/2.2.2/js/buttons.html5.min.js"></script>
<script src="https://cdn.datatables.net/buttons/2.2.2/js/buttons.print.min.js"></script>
```

Figure 100: Datatables CDN

DataTables is a jQuery JavaScript library plug-in. It's a powerful tool that adds instant search, pagination and export features which provides a fully interactive table to websites.

## 18.6 Customer Module

### 18.6.1 Create – HTML Form, PHP

#### User Contact-Us form

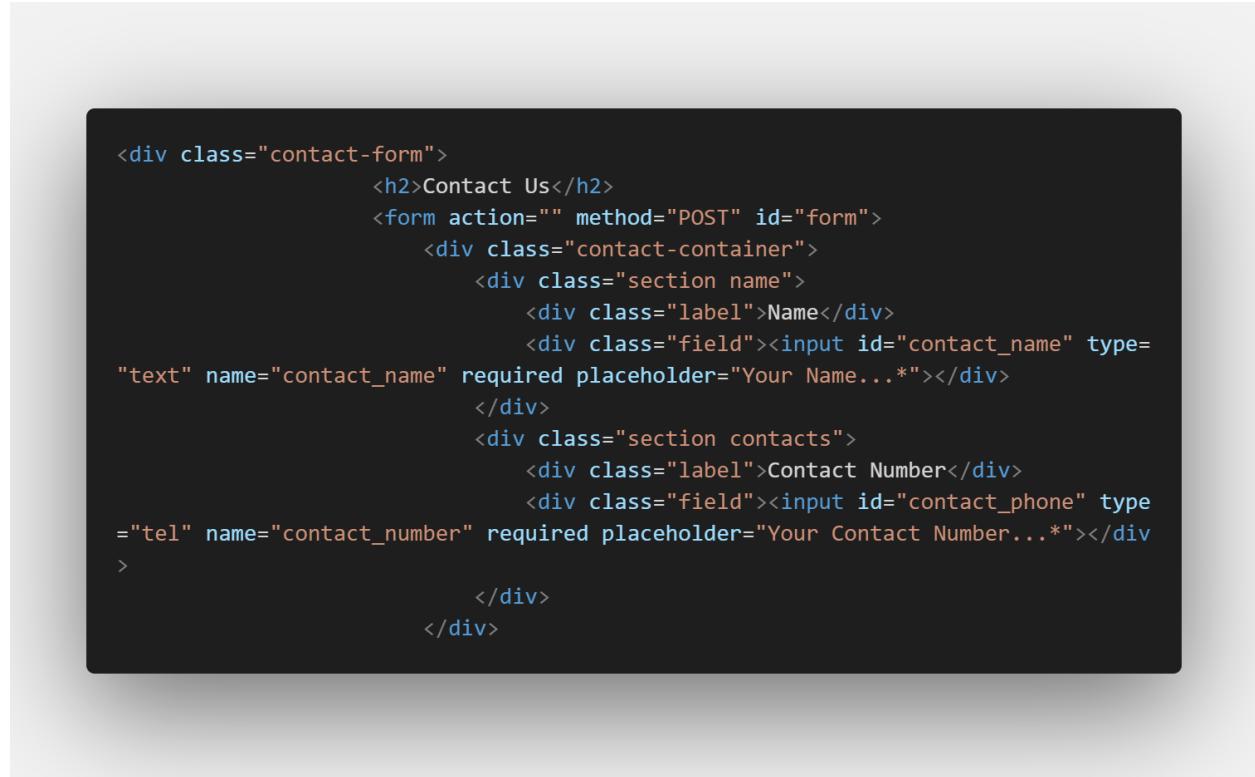


Figure 101 Contact-us form

One of the many features provided for the users includes the contact us function where the user can opt to send messages which includes suggestion, recommendation, questions or feedbacks, all of which in the form of text. Starting off with the form is located in contact\_us.html which consist of multiple lines of source code which provides information on the contact details, social links and headings. Aside from that, the webpage also includes a contact us form which is accessible by the user in the contact us page. With the use of a form tag, users are given multiple fields labeled by the name of the required input, for example, in the name field, the input is labelled as “Name” via a division or div tag. Under the particular labels resides the input field where the user will interact with by providing their personal information. For example, for the input field for the user’s full name with the help of an input tag. Additionally, the input tag is set to their respective type attribute such as the contact name being in the form of a text while the user’s contact number is set to tel type for the respective type attribute. All the input tags also includes a required attribute

in the input tag which signifies the requirement where the input field are not to be left empty. This is also part of the validation process which an error will be prompt if the incorrect data type is entered or any fields with the attribute “required” are left empty. Additionally, for user’s readability, further enhancements are made to the form validation with the help of CSS where invalid input values will have a border of red while input values which are acceptable will show a green border around the input field.

```
<div class="section email">
    <div class="label">Email</div>
    <div class="field"><input id="contact_email" type="email" name="contact_email" required placeholder="Your Email Address...*"/></div>
    <div class="section comment">
        <div class="label">Comment</div>
        <div class="field"><textarea name="comment" placeholder="Comment...*" required></textarea></div>
    </div>
    <div class="section">
        <div class="label">&nbsp;</div>
        <div class="field">
            <button type="reset" class="btn-control">Reset</button>
            <button type="submit" onclick="return confirm('Are you sure you wish to submit this comment?')" class="btn-control">Submit</button>
        </div>
    </div>
</form>
</div>
```

Figure 102 User Contact Us form (cont')

## Book Appointments

```
<div>
    <form action="confirm_wth_payment.php" method="POST" id="form1">
        <fieldset class="fieldset">
            <div class="hstack gap-3 mb-3">
                <div>
                    <label for="name" class="form-label">Full Name:</label>
                    <input class="form-control fullName" type="text" name="cus_name" id="input-name"
placeholder="John Doe" value="<?php echo $patientName?>" required>
                </div>
                <div>
                    <label for="contact_number" class="form-label">Contact Number:</label>
                    <input class="form-control contactNum" type="tel" name="cus_tel" id="input-tel"
placeholder="000-000-0000" required>
                </div>
            </div>
            <div class="mb-3">
                <label for="doctor_available">Doctor Available:</label>
                <select name="doctor" id="doctor" class="form-select doctorTreatment" required>
                    <option>Select a Doctor</option>
                    <?php include("conn.php");
                        //Get list of doctors available
                        $doctorList = mysqli_query($con, "SELECT * FROM tbl_doctor;");
                    <?>
                    <?php
                        while ($doctorResult = mysqli_fetch_array($doctorList)) {
                            $doctor_available_list = '<option value=' . $doctorResult["DocID"] .
'">' . $doctorResult["Name"] . '</option>';
                            echo $doctor_available_list;
                        }
                    <?>
                </select>
            </div>
            <div>
                <label for="date_appointment">Pick an appointment date:</label>
                <?php
                    $todayDate = date("Y-m-d");
                    $nextDay_timestamp = strtotime("+1 day");
                    $nextDay = date("Y-m-d", $nextDay_timestamp);
                    $minDate = date("Y-m-d", strtotime("+1 day"));
                <?>
                    <input type="date" min="<?php echo $minDate?>" class="form-control dateAppoint"
name="appt_date" id="appt_date" required>
                </div>
                <div class="mt-3 d-flex justify-content-end">
                    <button type="submit" class="btn btn-secondary next-btn" id="flag-
next">Next</button>
                    <button type="submit" formaction="check_slots.php" class="btn btn-primary checkBtn"
id="checkBtn" hidden></button>
                </div>
                <div id="sample"></div>
            </fieldset>
        <!-- </form> -->
    </div>
</div>
```

Figure 103 Book Appointment Form



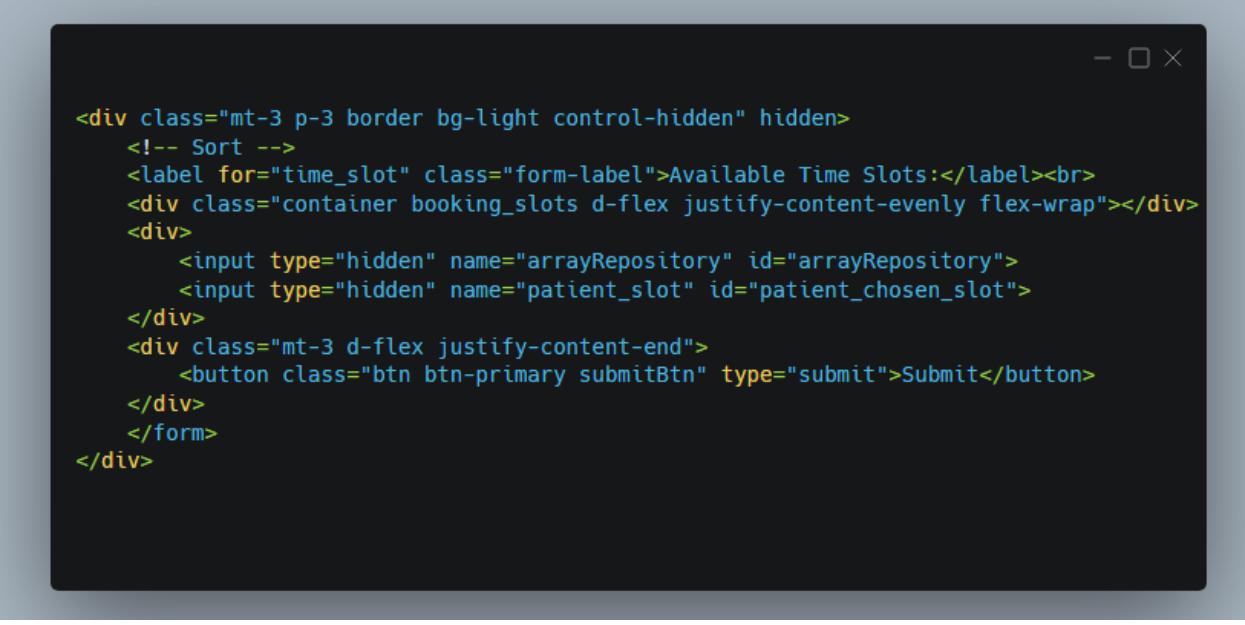
Figure 104 Book Appointment Form (cont')

Similarly, to the input form mentioned in contact us section, the appointment booking also includes a form which shown in Figure 102, having an id of form1, method POST and form action to which the form will be submitted to confirm\_wth\_payment.php. Similar to the form provided in contact us section, booking appointment also consist of label tags as well as input tags which provide user interaction all of while performing booking functionality. For example, the user are required to enter their personal information such as their full name, contact number, doctors which they wish to consult, appointment date as well as slots for their appointment. The form also includes various validation methods where the data type plays an important role in form validation as well as the input values being entered. For instance, one of the component in the booking process which is the selection of the appointment date and time slots, the users are not allowed to enter date from the past as well as time slots which are already booked by other users. All of which are filtered with the help of PHP and JavaScript. For instance, in order to perform validation on the input type for appointment date, the min attribute is added into the input field for appointment date which disables invalid date entries. Aside from that, input validation for the selection of appointment slots are also provided by disabling any booked appointment on the selected date which denies user from booking the same slot as other users (Figure 105).



```
<div>
    <label for="date_appointment">Pick an appointment date:</label>
    <?php
        $todayDate = date("Y-m-d");
        $nextDay_timestamp = strtotime("+1 day");
        $nextDay = date("Y-m-d", $nextDay_timestamp);
        $minDate = date("Y-m-d", strtotime("+1 day"));
    ?>
    <input type="date" min="<?php echo $minDate;?>" class="form-control dateAppoint" name="appt_date"
    id="appt_date" required>
</div>
```

Figure 105 Form validation for input type: date



```
<div class="mt-3 p-3 border bg-light control-hidden" hidden>
    <!-- Sort -->
    <label for="time_slot" class="form-label">Available Time Slots:</label><br>
    <div class="container booking_slots d-flex justify-content-evenly flex-wrap"></div>
    <div>
        <input type="hidden" name="arrayRepository" id="arrayRepository">
        <input type="hidden" name="patient_slot" id="patient_chosen_slot">
    </div>
    <div class="mt-3 d-flex justify-content-end">
        <button class="btn btn-primary submitBtn" type="submit">Submit</button>
    </div>
    </form>
</div>
```

Figure 106 Display Available Slots HTML

## Insert User Bookings into Database



```
<?php
    include('session.php');
    include('conn.php');

    if(isset($_POST['newSlot'])) {
        $newSlot = str_replace(".", ":", $_POST['newSlot']);
        $newSlot .= ":00";
    }
    echo $newSlot, $Acc_ID, $_POST["doctor"], $_POST["appt_date"], $_POST["appStatus"];

    $sqlInsertAppt = "INSERT INTO tbl_appointment (Acc_ID, DocID, App_date, App_slot, App_status) values
    ('$Acc_ID', '$_POST[doctor]', '$_POST[appt_date]', '$newSlot', '$_POST[appStatus]');";

    if (!mysqli_query($con, $sqlInsertAppt)){
        echo "APPT error";
        die('Error: ' . mysqli_error($con));
    } else {
        $resultNewAppt = mysqli_query($con, "SELECT * FROM tbl_appointment WHERE (Acc_ID ='$Acc_ID') AND
        (App_date = '$_POST[appt_date]' ) AND (App_slot = '$newSlot')");
        while ($row = mysqli_fetch_array($resultNewAppt)) {
            $AppID = $row['AppID'];
        }
    }

    $paymentDate = date("Y-m-d");
    echo $Acc_ID, $AppID, $_POST["paymentTotal"], $_POST["paymentMethod"], $paymentDate,
    $_POST["paymentStatus"];

    $sqlInsertPayment = "INSERT INTO tbl_payment (Acc_ID, AppID, Payment_total, Payment_method,
    Payment_date, Payment_status) values
    ('$Acc_ID', '$AppID', '$_POST[paymentTotal]', '$_POST[paymentMethod]', '$paymentDate',
    '$_POST[paymentStatus]')";
    if (!mysqli_query($con, $sqlInsertPayment)){
        echo "Payment error";
        die('Error: ' . mysqli_error($con));
    } else {
        echo '<script>alert("Your Bookings have been added!");';
        window.location.href= "book_treatment.php";
        '</script>';
    }

    mysqli_close($con);
?>
```

Figure 107 confirm\_payment.php (SQL)

Once the user had successfully entered their respective information as well as booking details, they will be requested to choose a payment method where the user can opt to pay later or pay now by cash or by online banking. After which the user had made their selection, the system will create an entry into valleyclinic database by inserting the provided information such as user's appointment date, their associated account, doctor information, and their chosen slot. Once the information are successfully store into `tbl_appointment`, user will be redirected back to the booking page. Additionally, the user's preferable choice of payment method will also be recorded via

inserting the information that was previously obtained in patient\_payment.php such as the payment total, payment method of choice, and the payment date.

### 18.6.2 Login – HTML, Form, PHP

```
<div class="container">
  <div class="header">
    <h2>Sign In</h2>
    <div class="error">
      <?php if (isset($_GET['error'])) {
        ?>
        <p class="error"><?php echo $_GET['error']; ?></p>
      <?php
      } ?>
    </div>
  </div>

  <form method="POST" class="form" action="logincheck.php">
    <div class="form-control">
      Username:
      <input type="text" name="username" placeholder="Username" autocomplete="off">
    </div>
    <div class="form-control">
      Password:
      <input type="password" name="password" placeholder="Password" autocomplete="off">
    </div>
    <button name="submit">Sign In</button>
    <div class="signup_link">
      Not a member? <a href="signup.php">Click here to Sign Up!</a>
    </div>
  </form>
</div>
</div>
```

```

if (empty($username) && empty($pass))
{
    header("Location: signin.php?error=Fields cannot be empty!");
    exit();
}
else if (empty($username))
{
    header("Location: signin.php?error=Username is required!");
    exit();
}
else if(empty($pass))
{
    header("Location: signin.php?error=Password is required!");
    exit();
}
else
{
    $pass = md5($pass);
    $sql = "SELECT * FROM users WHERE Username = '$username' AND Password = '$pass'";

    $result = mysqli_query($conn, $sql);

    if (mysqli_num_rows($result) === 1)
    {
        $row = mysqli_fetch_assoc($result);

        if ($row['Username'] === $username && $row['Password'] === $pass)
        {
            $_SESSION['Username'] = $row['Username'];
            $_SESSION['Acc_ID'] = $row['Acc_ID'];
            header("Location: account.php");
        }
        else
        {
            header("Location: signin.php?error= Incorrect Username or Password!");
            exit();
        }
    }
    else
    {
        header("Location: signin.php?error= Incorrect Username or Password!");
        exit();
    }
}

```

Figure 108 confirm\_payment.php (SQL)

Based on the first picture, when the user clicks the sign-in button on the form it will execute the codes based on the second picture. The username and password will be stored in a variable and then later be used for later parts of the code. On the IF statement where (empty(\$username) && empty(\$password)) means that if the fields of the username and password is empty it will execute the code and the code where “header(“Location: signin.php?error=Fields cannot be empty!&userdata”)” it will display that message on top of the form. As for the rest of the else if statements it will be the same concept as explained. If the user has entered everything into the form, the else statement will be executed. It will read the data from the form that the user has inputted and then it will verify the data into from database table where the code “if (\$row['Username'] === \$username && \$row['Password'] === \$pass)

```
{  
    $_SESSION['Username'] = $row['Username'];  
    $_SESSION['Acc_ID'] = $row['Acc_ID'];  
    header("Location: account.php");  
}
```

)". Once the code has been executed, it will redirect the user to the homepage.

### 18.6.3 Signup – HTML, Form, PHP

```
<form method="POST" class="form" action="signupcheck.php" >
<div class="form-control">
    Username:
    <?php if (isset($_GET['username'])) {
        ?>
        <input type="text" name="username" placeholder="Username" value="<?php echo $_GET['username']; ?>">
    </?php
    } else {
        ?>
        <input type="text" name="username" placeholder="Username" autocomplete="off">
    </?php
    }?>
</div>
<div class="form-control">
    Password:
    <?php if (isset($_GET['password'])) {
        ?>
        <input type="password" name="password" placeholder="Password" value="<?php echo $_GET['password']; ?>">
    </?php
    } else {
        ?>
        <input type="password" name="password" placeholder="Password" autocomplete="off">
    </?php
    }?>
</div>
<div class="form-control">
    Confirm Password:
    <?php if (isset($_GET['password2'])) {
        ?>
        <input type="password" name="password2" placeholder="Confirm Password" value="<?php echo $_GET['password2']; ?>">
    </?php
    } else {
        ?>
        <input type="password" name="password2" placeholder="Confirm Password" autocomplete="off">
    </?php
    }?>
</div>
<button name="submit">Sign Up</button>
<div class="signup_link">
    Already a member? <a href="signin.php">Click here to Sign In!</a>
</div>
</form>
```

```
<?php
session_start();
include "conn.php";

if (isset($_POST['username']) && isset($_POST['password']) && isset($_POST['password2']))
{
    function validate($data)
    {
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
    }

    $username = validate($_POST['username']);
    $pass = validate($_POST['password']);
    $pass2 = validate($_POST['password2']);

    $userdata = 'username=' . $username;

    if (empty($username) && empty($pass) && empty($pass2))
    {
        header("Location: signup.php?error=Fields cannot be empty!&$userdata");
        exit();
    }
    else if(empty($username))
    {
        header("Location: signup.php?error=Username is required!&$userdata");
        exit();
    }
    else if(empty($pass))
    {
        header("Location: signup.php?error=Password is required!&$userdata");
        exit();
    }
    else if(empty($pass2))
    {
        header("Location: signup.php?error=Confirmation password is required!&$userdata");
        exit();
    }
    else if($pass != $pass2)
    {
        header("Location: signup.php?error=Passwords do not match!&$userdata");
        exit();
    }
    else
    {
```

```

else
{
    $pass = md5($pass);
    $sql = "SELECT * FROM users WHERE Username = '$username'";

    $result = mysqli_query($conn, $sql);

    if (mysqli_num_rows($result) > 0){
        header("Location: signup.php?error=Username has been taken!&$userdata");
        exit();
    }
    else
    {
        $sql2 = "INSERT INTO users(Username, Password) VALUES('$username', '$pass')";
        $result2 = mysqli_query($conn, $sql2);
        if ($result2)
        {
            header("Location: signin.php");
            exit();
        }
        else
        {
            header("Location: signup.php?error=Unknown error occurred!&$userdata");
            exit();
        }
    }
}
else
{
    header("Location: signup.php");
    exit();
}

```

Based on the first picture, when the user clicks the sign-up button on the form it will execute the codes based on the second picture. The username, password, confirmation password will be stored in a variable and then later be used for later parts of the code. On the IF statement where (empty(\$username) && empty(\$password) && empty(\$password2)) means that if the fields of the username, password, confirmation password is empty it will execute the code and the code where “header(“Location: signup.php?error=Fields cannot be empty!&userdata”)” it will display that message on top of the form. As for the rest of the else if statements it will be the same concept as explained. If the user has entered everything into the form, the code in the third picture will be executed. It will read the data from the form that the user has inputted and then it will insert the data into the database table where the code “\$sql2= “INSERT INTO users(Username, Password) VALUES(‘\$username’,’\$pass’)” it will insert into the database table. Once the code has been executed, it will redirect the user to the sign-in page.

## 18.6.4 Update – HTML, Form, PHP

```
<?php
<form method="POST" class="form" action="updateaccount.php" >
    <?php
        $current = $_SESSION['Username'];
        $sql = "SELECT * FROM users WHERE Username = '$current'";
        $results = mysqli_query($conn,$sql);

        if($results)
        {
            if(mysqli_num_rows($results)>0)
            {
                while($row = mysqli_fetch_array($results))
                {
                    <?>
                    <div class="form-control">
                        Username:
                        <input type="text" name="username" value = "<?php echo $row['Username'];??" >
                    </div>
                    <div class="form-control">
                        Password:
                        <input type="password" name="password" value = "<?php echo $row['Password'];??" >
                    </div>
                    <div class="form-control">
                        Name:
                        <input type="text" name="name" value = "<?php echo $row['Name'];??" >
                    </div>
                    <div class="form-control">
                        Phone Number:
                        <input type="text" name="phone" value = "<?php echo $row['Phone'];??" >
                    </div>
                    <div class="form-control">
                        Email:
                        <input type="text" name="email" value = "<?php echo $row['Email'];??" >
                    </div>
                    <button name="submit">Save</button>
                <?php
            }
        }
    <?>
</form>
```

```
<?php
session_start();
include 'conn.php';

if(isset($_POST['submit']))
{
    $newUser = $_POST['username'];
    $newPass = $_POST['password'];
    $newName = $_POST['name'];
    $newPhone = $_POST['phone'];
    $newEmail = $_POST['email'];

    if(empty($newUser) && empty($newPass) && empty($newName) && empty($newPhone) && empty($newEmail))
    {
        $loggedinuser = $_SESSION['Username'];
        $sql = "UPDATE users SET Username = '$newUser', Password = '$newPass', Name = '$newName', Phone = '$newPhone', Email = '$newEmail' WHERE Username = '$loggedinuser'";
        $result = mysqli_query($conn, $sql);
        header("Location: account.php?success=Successfully Updated Account Details!");
        exit();
    }
    else
    {
        header("Location: account.php?error=Fields cannot be empty!");
        exit();
    }
}
```

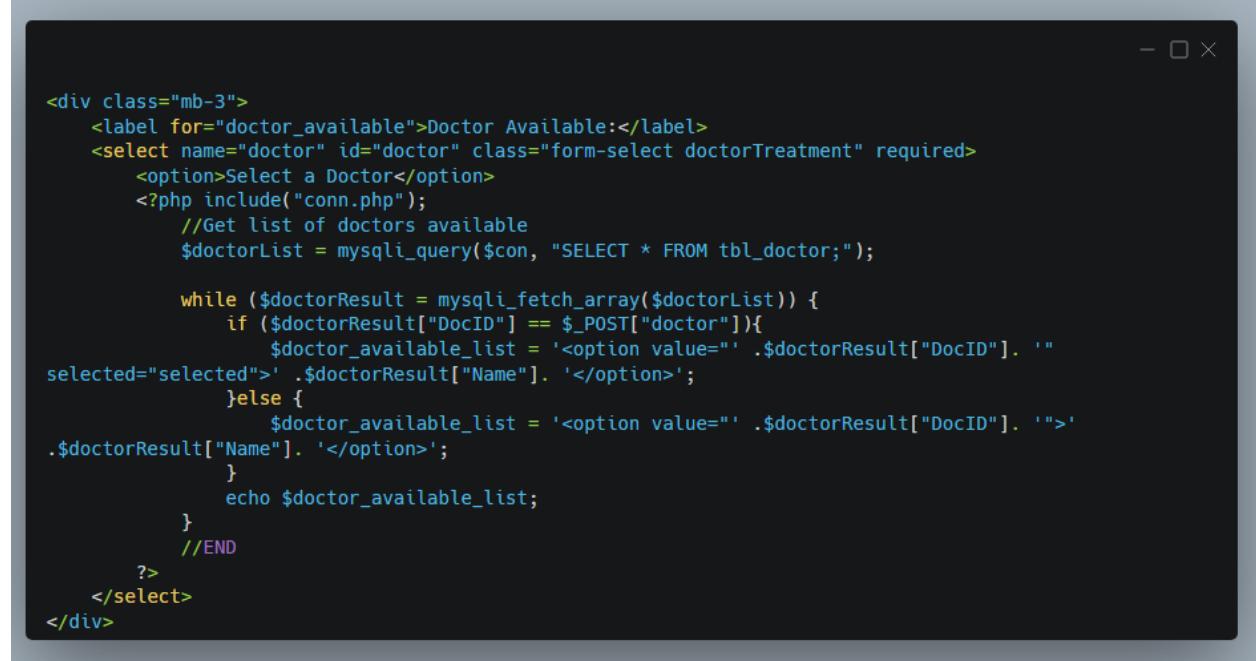
Based on the first picture, the user data will be displayed in the form and will get the data from the database based on the session and ID of the user's account. If the user decides to make changes to their account, for example their name or email, they will need to backspace their current data displayed on the form and then enter their new credentials. Once entered the user will need to click on the “Save” button. The form will then execute the codes based on the second picture as shown, the data will be stored in the variables and then the code “\$sql=”UPDATE users SET Username =

'\$newUser', Password = '\$newPass', Name = '\$newName', Phone = '\$newPhone', Email = '\$newEmail' WHERE Username = '\$loggedinuser'" it will replace the old data with the new data that the user has keyed in with the update function.

## User Confirm Booking Details



```
<form action="patient_payment.php" method="POST" id="final_submit">
    <div class="hstack gap-3 mb-3">
        <div>
            <label for="name" class="form-label">Full Name:</label>
            <input class="form-control fullName" type="text" name="cus_name" id="input-name"
placeholder="John Doe" value=<?php echo $_POST['cus_name']?>" required>
        </div>
        <div>
            <label for="contact_number" class="form-label">Contact Number:</label>
            <input class="form-control contactNum" type="tel" name="cus_tel" id="input-tel"
placeholder="000-000-0000" value=<?php echo $_POST['cus_tel']?>" required>
        </div>
    </div>
```



```
<div class="mb-3">
    <label for="doctor_available">Doctor Available:</label>
    <select name="doctor" id="doctor" class="form-select doctorTreatment" required>
        <option>Select a Doctor</option>
        <?php include("conn.php");
        //Get list of doctors available
        $doctorList = mysqli_query($con, "SELECT * FROM tbl_doctor");>

        while ($doctorResult = mysqli_fetch_array($doctorList)) {
            if ($doctorResult["DocID"] == $_POST["doctor"]){
                $doctor_available_list = '<option value="' . $doctorResult["DocID"]. '"'
selected="selected">' . $doctorResult["Name"]. '</option>';
            }else {
                $doctor_available_list = '<option value="' . $doctorResult["DocID"]. '">' .
$doctorResult["Name"]. '</option>';
            }
            echo $doctor_available_list;
        }
        //END
    ?>
    </select>
</div>
```

Figure 109 User Update Booking Details



The screenshot shows a dark-themed browser window with the following HTML code:

```
<div class="mb-3">
    <label for="date_appointment">Pick an appointment date:</label>
    <input type="date" class="form-control dateAppoint" value="<?php echo $_POST['appt_date']?>" name="appt_date" id="appt_date" required disabled>
    <input type="hidden" value="<?php echo $_POST['appt_date'];?>" id="chosenDate">
</div>
<div>
    <label for="time_slot" class="form-label">Available Time Slots:</label><br>
    <input id="chosen_slot" type="hidden" value="<?php echo $_POST['patient_slot']?>">
    <input type="hidden" id="newSlot" name="newSlot">
    <div class="container booking_slots d-flex justify-content-evenly flex-wrap" id="booking_slots">
</div>
    <div class="mt-3 d-flex justify-content-end gap-2">
        <button type="button" id="disAtt" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#staticBackdrop">Confirm Booking</button>
    </div>
</div>
```

Figure 110 User Update Booking Details (cont)

After which the user had submitted their booking details via the submit form button, the user will be redirected to confirm\_wth\_payment.php which allows the user to once again review the provided information whether the input are as desired to the user. Similar fashion are applied in the form section of confirm\_wth\_payment.php which the difference being, user's information that was previously entered are not echoed into the value attribute of each input. For example, user had selected their desired doctor to visit, in the selection menu in confirm\_wth\_payment.php, the user's options are not highlighted by applying the attribute selected in the option tag if the "IF" condition are met such that if the provided doctor ID is similar to the DocID in the database, the respective option tag will be selected. (Figure 108). Aside from the usual input tags as seen across multiple webpages, some input tags are also hidden to submit data across PHP files without being visible to the user, for example, the two input tags with the type as hidden are being used to transport data into patient\_payment.php such information includes the user's chosen appointment date as well as the time slot chosen as shown in Figure 109.

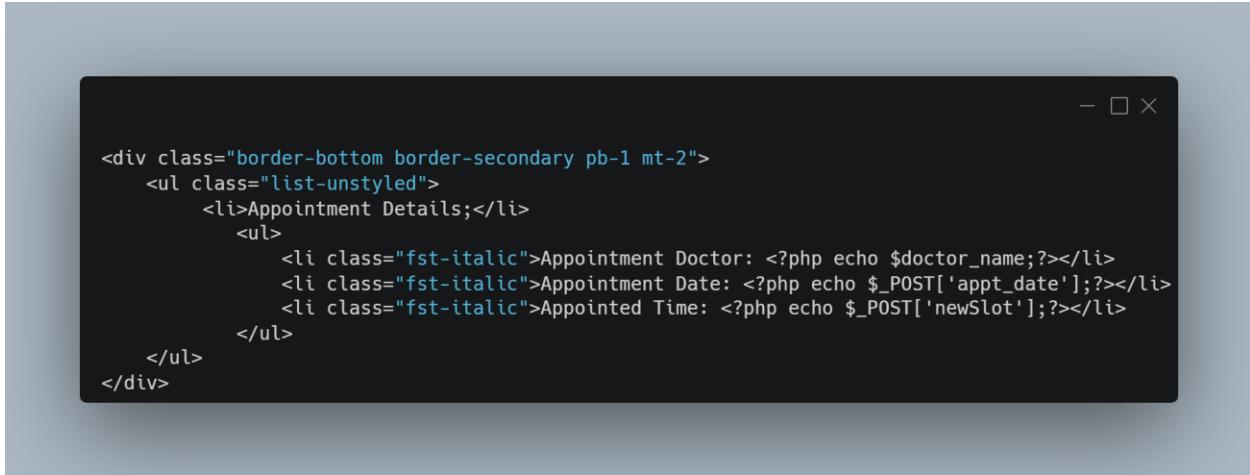
```
<div class="mb-3">
    <label for="doctor_available">Doctor Available:</label>
    <select name="doctor" id="doctor" class="form-select doctorTreatment" required>
        <option>Select a Doctor</option>
        <?php include("conn.php");
            //Get list of doctors available
            $doctorList = mysqli_query($con, "SELECT * FROM tbl_doctor;");

            while ($doctorResult = mysqli_fetch_array($doctorList)) {
                if ($doctorResult["DocID"] == $_POST["doctor"]){
                    $doctor_available_list = '<option value="' . $doctorResult["DocID"]. '"'
selected="selected">' . $doctorResult["Name"]. '</option>';
                }else {
                    $doctor_available_list = '<option value="' . $doctorResult["DocID"]. '">' .
$doctorResult["Name"]. '</option>';
                }
                echo $doctor_available_list;
            }
            //END
        ?>
    </select>
</div>
```

Figure 111 User selected doctor of choice

## 18.6.5 Read – HTML, Form, PHP

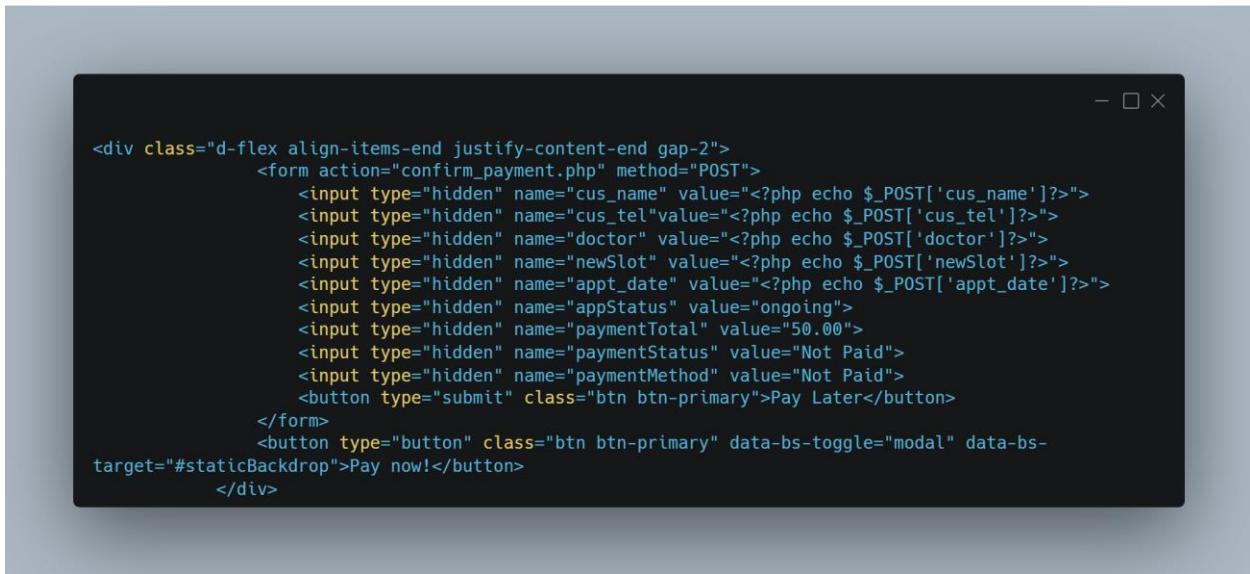
### User Payment



A screenshot of a web browser window showing a list of appointment details. The code in the screenshot is:

```
<div class="border-bottom border-secondary pb-1 mt-2">
    <ul class="list-unstyled">
        <li>Appointment Details;</li>
        <ul>
            <li class="fst-italic">Appointment Doctor: <?php echo $doctor_name;?></li>
            <li class="fst-italic">Appointment Date: <?php echo $_POST['appt_date'];?></li>
            <li class="fst-italic">Appointed Time: <?php echo $_POST['newSlot'];?></li>
        </ul>
    </ul>
</div>
```

Figure 112 User Booking Details for display



A screenshot of a web browser window showing a payment form. The code in the screenshot is:

```
<div class="d-flex align-items-end justify-content-end gap-2">
    <form action="confirm_payment.php" method="POST">
        <input type="hidden" name="cus_name" value="<?php echo $_POST['cus_name']?>">
        <input type="hidden" name="cus_tel" value="<?php echo $_POST['cus_tel']?>">
        <input type="hidden" name="doctor" value="<?php echo $_POST['doctor']?>">
        <input type="hidden" name="newSlot" value="<?php echo $_POST['newSlot']?>">
        <input type="hidden" name="appt_date" value="<?php echo $_POST['appt_date']?>">
        <input type="hidden" name="appStatus" value="ongoing">
        <input type="hidden" name="paymentTotal" value="50.00">
        <input type="hidden" name="paymentStatus" value="Not Paid">
        <input type="hidden" name="paymentMethod" value="Not Paid">
        <button type="submit" class="btn btn-primary">Pay Later</button>
    </form>
    <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#staticBackdrop">Pay now!</button>
</div>
```

Figure 113 Payment form for transfer user info (Pay later)

From the continuation of confirm\_wth\_payment.php, user will be redirected to patient\_payment.php upon form submission which includes the submission of booking details along with user's personal information. For example, in figure 106 and Figure 107 shown, user's personal information and their respective chosen appointment details are put into a form to be submitted an insert query where all the information will be stored into a database.

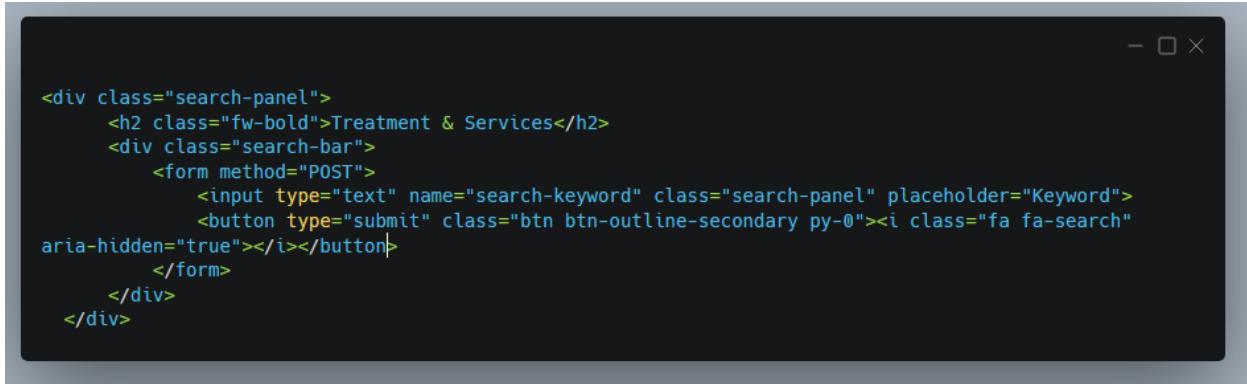
```
<div class="d-flex justify-content-center">
    <input type="hidden" name="cus_name" value="<?php echo $_POST['cus_name']?>">
    <input type="hidden" name="cus_tel" value="<?php echo $_POST['cus_tel']?>">
    <input type="hidden" name="doctor" value="<?php echo $_POST['doctor']?>">
    <input type="hidden" name="newSlot" value="<?php echo $_POST['newSlot']?>">
    <input type="hidden" name="appt_date" value="<?php echo $_POST['appt_date']?>">
    <input type="hidden" name="appStatus" value="ongoing">
    <input type="hidden" name="paymentTotal" value="50.00">
    <input type="hidden" name="paymentMethod" value="CreditCard">
    <input type="hidden" name="paymentStatus" value="Paid">
    <button class="btn btn-primary" type="submit">Proceed to Checkout</button>
</div>
```

Figure 114 Reading user input into a form (Credit Card method)

```
<div class="d-flex justify-content-center">
    <input type="hidden" name="cus_name" value="<?php echo $_POST['cus_name']?>">
    <input type="hidden" name="cus_tel" value="<?php echo $_POST['cus_tel']?>">
    <input type="hidden" name="doctor" value="<?php echo $_POST['doctor']?>">
    <input type="hidden" name="newSlot" value="<?php echo $_POST['newSlot']?>">
    <input type="hidden" name="appt_date" value="<?php echo $_POST['appt_date']?>">
    <input type="hidden" name="appStatus" value="ongoing">
    <input type="hidden" name="paymentTotal" value="50.00">
    <input type="hidden" name="paymentMethod" value="cash">
    <input type="hidden" name="paymentStatus" value="Not Paid">
    <button class="btn btn-primary" type="submit">Proceed to Checkout</button>
</div>
```

Figure 115 Reading user input into a form (Cash Method)

## Search Treatment and Services



A screenshot of a search bar interface. At the top, there is a header with the text "Treatment & Services". Below the header is a search bar containing an input field with the placeholder "Keyword" and a search button with a magnifying glass icon. The entire search panel is enclosed in a "search-panel" class.

```
<div class="search-panel">
    <h2 class="fw-bold">Treatment & Services</h2>
    <div class="search-bar">
        <form method="POST">
            <input type="text" name="search-keyword" class="search-panel" placeholder="Keyword">
            <button type="submit" class="btn btn-outline-secondary py-0"><i class="fa fa-search" aria-hidden="true"></i></button>
        </form>
    </div>
</div>
```

Figure 116 Search bar for Treatment and Services



A screenshot of a PHP script. The script starts with a standard PHP opening tag. It includes a file named "conn.php" and then queries a database for treatments. If no results are found, it displays a message indicating that there is no such treatment available and encourages users to contact staff. If results are found, it loops through each row, checks if there is a picture, and creates a thumbnail gallery entry for each treatment, including the image source, alt text, and a link to the service details page. Finally, it prints the generated HTML and closes the database connection.

```
<div class="main-content">
<?php
    include("conn.php");
    $result_treatments = mysqli_query($con, $sql_treatment);

    if (mysqli_num_rows($result_treatments) == 0 ) {
        echo '<div class="ts_gallery" style="border: none; margin: 0 auto; width: 100%;"><p>There is no such treatment available!<br>Please contact our staff for more information.</p></div>';
    }

    while($row = mysqli_fetch_array($result_treatments)) {
        if ($row['Picture'] == "") {
            $treatment_pic = "img-placeholder.png";
        } else {
            $treatment_pic = $row['Picture'];
        }

        $to_print = '<div class="ts_gallery">
            
            <a href="medical_services.php?ServID=' . $row['ServID']. '">' . $row['Title']. '</a>
        </div>';

        echo $to_print;
    }
    mysqli_close($con);
?
</div>
```

Figure 117 Search Treatment and Service PHP

As shown in Figure 111 and Figure 112, depicts the source code used in treatment and services webpage where the user can opt to search for relevant medical conditions and treatments based on the provided keyword. On top of the search feature, the user will be provided with an input field where the user are to key in keyword or alphabetical letters into the input field. Upon

submission of the provided keyword, a query statement will be executed using the keyword as a condition checked with the stored medical condition name. If the selected result from the executed mysql query resulted in no result, an error message will be prompt to notify the user of no such result matches the database, whereas if there exist more than one search result, a loop will be executed to print out the information in the form of a gallery.



A screenshot of a code editor window displaying a PHP script for generating pagination. The code includes logic for connecting to a MySQL database, searching for medical services by keyword, calculating the number of pages, and generating an HTML ul element with li elements for each page, including previous and next links.

```
<?php
    include("conn.php");
    $result_per_page = 8;
    $search_key = "";

    if(isset($_POST['search-keyword'])) {
        $search_key = $_POST['search-keyword'];
    } else {
        $search_key = "";
    }
    $sql_treatment = "SELECT * FROM tbl_medical_services WHERE Title LIKE '%$search_key%'";
    $result_row = mysqli_query($con, $sql_treatment);
    $num_rows = mysqli_num_rows($result_row);

    $numPages = ceil($num_rows / $result_per_page);

    //Current Page
    if (!isset($_GET['page'])) {
        $page = 1;
    } else {
        $page = $_GET['page'];
    }

    $starting_limit = ($page - 1) * $result_per_page;

    // Display pagination
    echo '<div class="pagination">';
    echo '<ul class="pagination-menu">';
    if ($page > 1) {
        echo '<li class="nav-prev"><a href="treatmentNservices.php?page=' . ($page - 1) . '">&laquo;/a>
    </li>';
    }
    for ($i = 1; $i <= $numPages; $i++) {
        echo '<li class="nav-numb"><a href="treatmentNservices.php?page=' . $i . '">' . $i . '</a></li>
    ';
    }
    if ($i > $page) {
        echo '<li class="nav-next"><a href="treatmentNservices.php?page=' . ($page + 1) . '">&raquo;/a>
    </li>';
    }
    echo '</ul>';
    echo '</div>';

    mysqli_close($con);
?>
```

Figure 118 Pagination PHP

In addition to the search feature provided in treatment and services webpage, one of the many significant source code consist of the pagination feature provided for user interaction which is accessible in the treatment and services page. Upon user's visit, a pagination nav bar will be generated underneath the search result or the treatment and services list which allow the user to view up to 8 result per page. If there exist more than 8 results, then a new page will be used to view the rest of the information.

## View Doctor Information



```
<div class="body-container">
    <?php
        include("conn.php");
        $result_doctor = mysqli_query($con, "SELECT * FROM tbl_doctor");

        while($row = mysqli_fetch_array($result_doctor)) {

            if ($row['Picture'] == "") {
                $doctor_pic = "doctor-test.jpg";
            } else {
                $doctor_pic = $row['Picture'];
            }

            $to_print = '<div class="doctor-gallery">
                
                <h3>' . $row['Name']. '</h3>
                <h4>' . $row['Specialist_category']. '</h4>
            </div>';

            echo $to_print;
        }
        mysqli_close($con);
    ?>
</div>
```

Figure 119 Display Doctor Info (PHP)

Based on Figure 114 shown, the source code presented consist of the use of division tags as well as other HTML elements such as heading tags (h3 and h4) as well as other PHP elements. In our doctors webpage, users will be displayed with an array of galleries which consist of the images our doctors as well as their name and specialist. With the use of PHP, the source code uses the include function which execute the file named conn.php, mainly used for connecting the database. Aside from that, by executing the SELECT query, a while loop is used to iterate the array

fetch from mysqli\_fetch\_array. If there are any picture set in the database for that particular doctor, then the image will replace the default placeholder image, else a default image is set. Lastly, all the information from the database on the doctor's details will be printed out through the echo function.

### conn.php



```
<?php
$con=mysqli_connect("localhost","root","","valleyclinic");

//Check connection
if (mysqli_connect_error())
{
echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
?>
```

Figure 120 conn.php

Conn.php consist of the source code used in many of the PHP files in Valley Clinic webpage where numerous source code which are related to manipulation of data in the database are used. The PHP file above establish a connection with the database through the function mysqli\_connect which uses three parameters such as the host, username, password, and the database name. Aside from that a conditional statement are also in use for validation whether the connection with the database is established or else an error will be displayed.

```

if (empty($username) && empty($pass))
{
    header("Location: signin.php?error=Fields cannot be empty!");
    exit();
}
else if (empty($username))
{
    header("Location: signin.php?error=Username is required!");
    exit();
}
else if(empty($pass))
{
    header("Location: signin.php?error=Password is required!");
    exit();
}
else
{
    $pass = md5($pass);
    $sql = "SELECT * FROM users WHERE Username = '$username' AND Password = '$pass'";

    $result = mysqli_query($conn, $sql);

    if (mysqli_num_rows($result) === 1)
    {
        $row = mysqli_fetch_assoc($result);

        if ($row['Username'] === $username && $row['Password'] === $pass)
        {
            $_SESSION['Username'] = $row['Username'];
            $_SESSION['Acc_ID'] = $row['Acc_ID'];
            header("Location: account.php");
        }
        else
        {
            header("Location: signin.php?error= Incorrect Username or Password!");
            exit();
        }
    }
    else
    {
        header("Location: signin.php?error= Incorrect Username or Password!");
        exit();
    }
}

```

The username and password will be stored in a variable and then later be used for later parts of the code. On the IF statement where (empty(\$username) && empty(\$password)) means that if the fields of the username and password is empty it will execute the code and the code where “header(“Location: signin.php?error=Fields cannot be empty!&userdata”)” it will display that message on top of the form. As for the rest of the else if statements it will be the same concept as explained. If the user has entered everything into the form, the else statement will be executed. It will read the data from the form that the user has inputted and then it will verify the data into from database table where the code “if (\$row['Username'] === \$username && \$row['Password'] === \$pass)

```

{
    $_SESSION['Username'] = $row['Username'];

    $_SESSION['Acc_ID'] = $row['Acc_ID'];
}
```

```

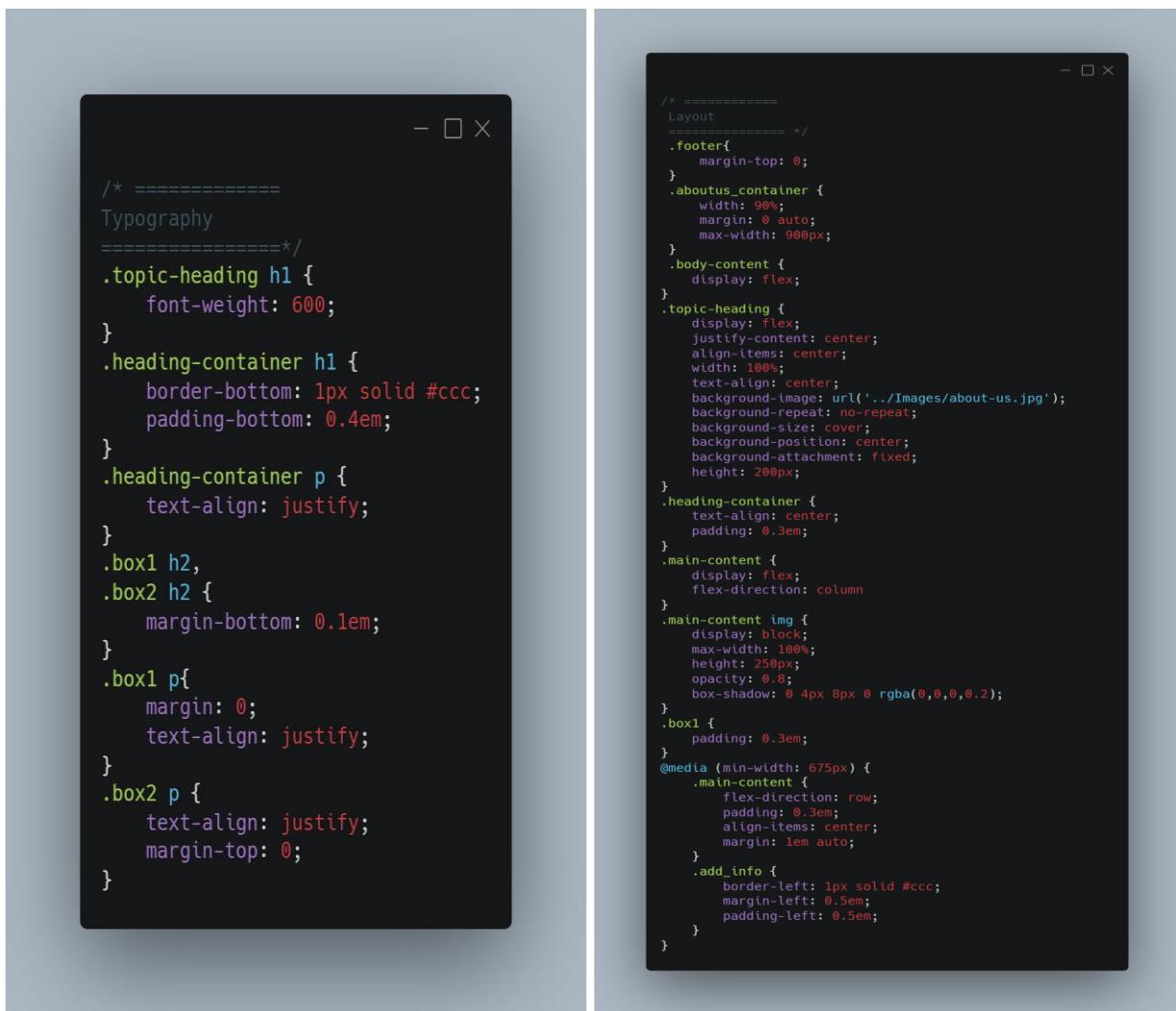
        header("Location: account.php");
    }

)");

```

### 18.6.6 Self-created CSS and JavaScript

#### About us CSS



```

/*
=====
Typography
=====
*/
.topic-heading h1 {
    font-weight: 600;
}
.heading-container h1 {
    border-bottom: 1px solid #ccc;
    padding-bottom: 0.4em;
}
.heading-container p {
    text-align: justify;
}
.box1 h2,
.box2 h2 {
    margin-bottom: 0.1em;
}
.box1 p{
    margin: 0;
    text-align: justify;
}
.box2 p {
    text-align: justify;
    margin-top: 0;
}

/* =====
Layout
=====
*/
.footer{
    margin-top: 0;
}
.aboutus_container {
    width: 90%;
    margin: 0 auto;
    max-width: 900px;
}
.body-content {
    display: flex;
}
.topic-heading {
    display: flex;
    justify-content: center;
    align-items: center;
    width: 100%;
    text-align: center;
    background-image: url('../Images/about-us.jpg');
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center;
    background-attachment: fixed;
    height: 200px;
}
.heading-container {
    text-align: center;
    padding: 0.3em;
}
.main-content {
    display: flex;
    flex-direction: column;
}
.main-content img {
    display: block;
    max-width: 100%;
    height: 250px;
    opacity: 0.8;
    box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
}
.box1 {
    padding: 0.3em;
}
@media (min-width: 675px) {
    .main-content {
        flex-direction: row;
        padding: 0.3em;
        align-items: center;
        margin: 1em auto;
    }
    .add_info {
        border-left: 1px solid #ccc;
        margin-left: 0.5em;
        padding-left: 0.5em;
    }
}

```

Figure 121 About Us CSS

Based on the Figure 114 shown above, the source code as shown in the diagram shows how each component of about us webpage are being styled. In the CSS page for the respective page, multiple CSS components are used which include the class and id selector as well as html selectors are also being used to style the respective element in the HTML file. Aside from that,

other attributes such as font-weight, paddings, text-alignment, margins, width, height, and etc. are some of the CSS attributes used in order to style the webpage. Additionally, one of the most significant piece of coding which consist of the flex grid which is being used throughout the webpages as shown in the system. All of which is provided through the use of display attribute with the value flex which allow for more room for customization in terms of object orders, alignment and justifications, as well as one of the major implementation in flex objects which are the width size sizing themselves based on the minimum width of which fits the content. Additionally, the CSS present also include the use of media query which allows for more room for customization where the screen design changes based on the style applied in the media query. For instance, if the user's screen size fall under (min-width) or above (max-width) of the condition in the media query, the block of codes inside the media query will be executed and the styles will be applied to their respective elements.

## Book Treatment CSS



```

/* =====
Typography
===== */
body {
    margin: 0 auto;
    font-family: 'Sora', sans-serif;
}
.about-us h3 {
    padding-bottom: 0.3em;
    border-bottom: 2px solid #f8f8f8;
}
.about-us p {
    text-align: justify;
}
.contact-us h3 {
    margin-bottom: 0;
    padding-bottom: 0.3em;
    border-bottom: 2px solid #f8f8f8;
}
.address i{
    font-size: 1.3rem;
    padding-top: 1.3em;
}
.phone-number i {
    font-size: 1.3.rem;
    padding-top: 1.2em;
}
.email i {
    padding-top: 0.1em;
}
.follow-us h3 {
    border-bottom: 2px solid #f8f8f8;
    padding-bottom: 0.3em;
    margin-bottom: 0;
}
a#fbLink {
    color: #1877F2;
}
a#instalink {
    color: #E4405F;
}
a#twitterLink {
    color: #1DA1F2;
}

```

```

/* =====
Layout
===== */
.container-bootstrap_control {
    width: 100%;
    margin: 0 auto;
    max-height: 120px;
    border-bottom: 25px solid navy;
}
.container-nav {
    margin-left: 0;
    display: flex;
    justify-content: space-between;
}

header {
    background-color: #f8f8f8;
}
@media (max-width: 675px) {
    .container-bootstrap_control {
        max-height: none;
    }
    .container-nav {
        flex-direction: column;
    }
    .header {
        text-align: center;
    }
}

```

The image shows two dark-themed code editors side-by-side, displaying CSS code for a 'Book Treatment' page. The left editor contains CSS for the footer, including styles for the footer itself, a footer box, follow-us sections, and media queries for widths of 675px and above. The right editor contains CSS for a heading wrap, which includes a before pseudo-element with a background image of 'booking-bg.jpg', and a booking heading with a font size of 0.625rem for screens up to 768px.

```

/* Footer */
.footer {
  display: flex;
  margin-top: 2.5em;
  flex-direction: column;
  padding: 0.5em;
  background-color: #A1A6B4;
}
.footer-box {
  display: flex;
  flex-direction: row;
}
.follow-us {
  display: flex;
  flex-direction: column;
}
.follow-box {
  display: flex;
  justify-content: left;
  margin: 0.5em;
}
@media (min-width: 675px) {
  div.footer {
    display: flex;
    flex-direction: row;
    justify-content: space-between;
  }
  .about-us {
    width: 30%;
  }
  .contact-us {
    width: 37%;
  }
  .follow-us {
    width: 30%;
  }
}

.heading-wrap {
  position: relative;
}
.heading-wrap::before {
  content: '';
  display: block;
  position: absolute;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  opacity: 0.6;
  background-image: url('../Images/booking-bg.jpg');
  background-repeat: no-repeat;
  background-position: 50% 0;
  background-size: cover;
}
.booking-heading {
  position: relative;
  display: flex;
  justify-content: center;
  height: 10em;
  align-items: center;
}
@media (max-width: 768px) {
  .sidebar-1 i {
    font-size: 0.625rem;
  }
}

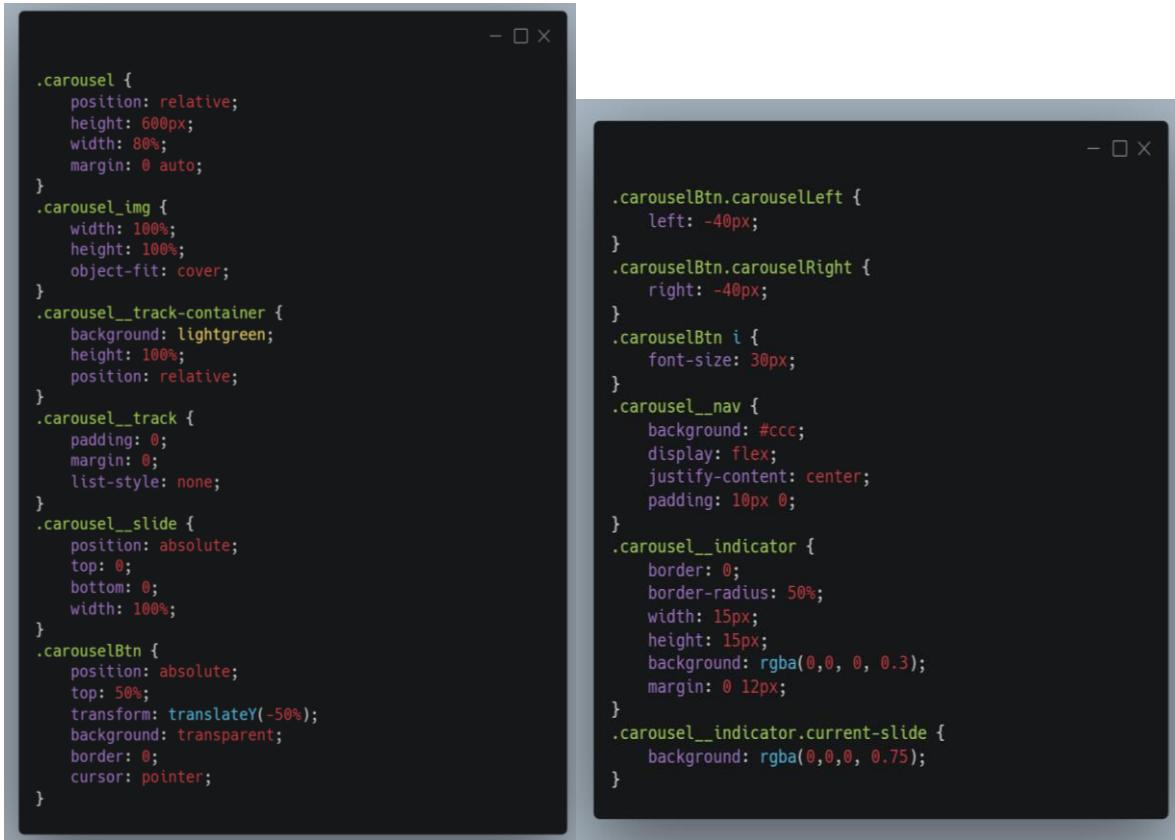
```

Figure 122 Book Treatment CSS

Based on the Figure 115 shown above, the source code as shown in the diagram shows how each component of bookings and appointment webpage are being styled. In the CSS page for the respective page, multiple CSS components are used which include the class and id selector as well as html selectors are also being used to style the respective element in the HTML file. Aside from that, other attributes such as font-weight, paddings, text-alignment, margins, width, height, and etc. are some of the CSS attributes used in order to style the webpage. Additionally, one of the most significant piece of coding which consist of the flex grid which is being used throughout the webpages as shown in the system. All of which is provided through the use of display attribute with the value flex which allow for more room for customization in terms of object orders, alignment and justifications, as well as one of the major implementation in flex objects which are the width size sizing themselves based on the minimum width of which fits the content. Additionally, the CSS present also include the use of media query which allows for more room for customization where the screen design changes based on the style applied in the media query. For instance, if the user's screen size fall under (min-width) or above (max-width) of the condition in

the media query, the block of codes inside the media query will be executed and the styles will be applied to their respective elements.

## Carousel CSS



```
.carousel {
    position: relative;
    height: 600px;
    width: 80%;
    margin: 0 auto;
}
.carousel_img {
    width: 100%;
    height: 100%;
    object-fit: cover;
}
.carousel__track-container {
    background: lightgreen;
    height: 100%;
    position: relative;
}
.carousel__track {
    padding: 0;
    margin: 0;
    list-style: none;
}
.carousel__slide {
    position: absolute;
    top: 0;
    bottom: 0;
    width: 100%;
}
.carouselBtn {
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
    background: transparent;
    border: 0;
    cursor: pointer;
}

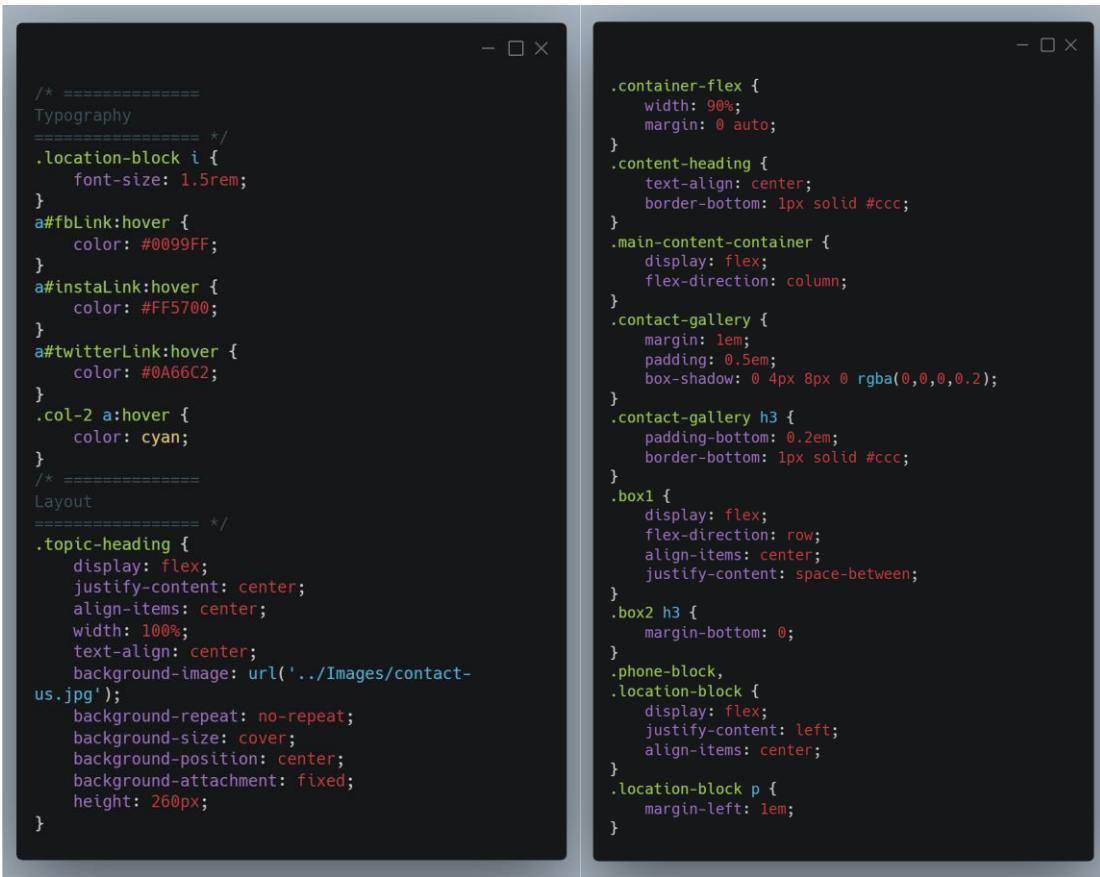
.carouselBtn.carouselLeft {
    left: -40px;
}
.carouselBtn.carouselRight {
    right: -40px;
}
.carouselBtn i {
    font-size: 30px;
}
.carousel__nav {
    background: #ccc;
    display: flex;
    justify-content: center;
    padding: 10px 0;
}
.carousel__indicator {
    border: 0;
    border-radius: 50%;
    width: 15px;
    height: 15px;
    background: rgba(0,0, 0, 0.3);
    margin: 0 12px;
}
.carousel__indicator.current-slide {
    background: rgba(0,0,0, 0.75);
}
```

Figure 123 Carousel CSS

Based on the Figure 116 shown above, the source code as shown in the diagram shows how the carousel component in homepage webpage are being styled. In the CSS page for the respective page, multiple CSS components are used which include the class and id selector as well as html selectors are also being used to style the respective element in the HTML file. Aside from that, other attributes such as font-weight, paddings, text-alignment, margins, width, height, and etc. are some of the CSS attributes used in order to style the webpage. Additionally, one of the most significant piece of coding which consist of the flex grid which is being used throughout the webpages as shown in the system. All of which is provided through the use of display attribute with the value flex which allow for more room for customization in terms of object orders, alignment and justifications, as well as one of the major implementation in flex objects which are

the width size sizing themselves based on the minimum width of which fits the content. Additionally, the CSS present also include the use of media query which allows for more room for customization where the screen design changes based on the style applied in the media query. For instance, if the user's screen size fall under (min-width) or above (max-width) of the condition in the media query, the block of codes inside the media query will be executed and the styles will be applied to their respective elements.

## Contact Us CSS



```
/* =====
Typography
=====
*/
.location-block i {
    font-size: 1.5rem;
}
a#fbLink:hover {
    color: #0099FF;
}
a#instaLink:hover {
    color: #FF5700;
}
a#twitterLink:hover {
    color: #0A66C2;
}
.col-2 a:hover {
    color: cyan;
}

/* =====
Layout
=====
*/
.topic-heading {
    display: flex;
    justify-content: center;
    align-items: center;
    width: 100%;
    text-align: center;
    background-image: url('../Images/contact-us.jpg');
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center;
    background-attachment: fixed;
    height: 260px;
}
```

```
.container-flex {
    width: 90%;
    margin: 0 auto;
}
.content-heading {
    text-align: center;
    border-bottom: 1px solid #ccc;
}
.main-content-container {
    display: flex;
    flex-direction: column;
}
.contact-gallery {
    margin: 1em;
    padding: 0.5em;
    box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
}
.contact-gallery h3 {
    padding-bottom: 0.2em;
    border-bottom: 1px solid #ccc;
}
.box1 {
    display: flex;
    flex-direction: row;
    align-items: center;
    justify-content: space-between;
}
.box2 h3 {
    margin-bottom: 0;
}
.phone-block,
.location-block {
    display: flex;
    justify-content: left;
    align-items: center;
}
.location-block p {
    margin-left: 1em;
}
```

```
@media (min-width: 675px) {
  .main-content-container {
    flex-direction: row;
    justify-content: space-evenly;
  }
  .contact-gallery {
    width: 30%;
  }
  .box1 a {
    display: block;
    margin: 1em;
  }
  .box1 i {
    font-size: 1.1rem;
    font-weight: bold;
  }
  .box1 {
    display: block;
    flex-direction: column;
  }
  .box3 h3 {
    margin-bottom: 0;
  }
}
.contact-form {
  display: flex;
  text-align: center;
  flex-direction: column;
  padding: 1em;
  box-sizing: border-box;
  box-shadow: 0 4px 8px 0 rgba(0,0,0,.2);
}
.contact-form h2 {
  border-bottom: 1px solid #ccc;
  margin-bottom: 0.3em;
  padding-bottom: 0.2em;
}
.contact-container {
  display: flex;
  justify-content: center;
}
.section {
  margin: 0.5em;
}
```

```
.section input,
.section textarea {
  border: 2px solid #f0f0f0;
  border-radius: 4px;
}
.section input:focus,
.section textarea:focus {
  outline: 0;
  border-color: #777;
}
.section textarea:valid,
.section input:valid {
  border-color: #2ecc71;
}
.section textarea:invalid,
.section input:invalid {
  border-color: #e74c3c;
}
.contacts,
.name {
  text-align: left;
}
.field input,
.field textarea {
  font-family: inherit;
  padding: 0.8em;
}
.email input {
  width: 71%;
}
.comment textarea {
  width: 71%;
  height: 100px;
}
.btn-control {
  background-color: #555555;
  color: white;
  display: inline;
  padding: 1rem 1.25rem;
  width: 49%;
  cursor: pointer;
  border: none;
  opacity: 0.9;
}
.btn:hover {
  opacity: 1;
```



```
@media (min-width: 675px) {
  .name input,
  .contacts input{
    width: 95%;
  }
  .contact-container {
    margin: 0 auto;
    width: 72%;
  }
  .name {
    width: 50%;
  }
  .contacts {
    width: 50%;
  }
}
```

Figure 124 Contact Us CSS

Based on the Figure 117 shown above, the source code as shown in the diagram shows how each component of contact us webpage are being styled such as the input form. In the CSS page for the respective page, multiple CSS components are used which include the class and id selector as well as html selectors are also being used to style the respective element in the HTML file. Aside from that, other attributes such as font-weight, paddings, text-alignment, margins, width, height, and etc. are some of the CSS attributes used in order to style the webpage. Additionally, one of the most significant piece of coding which consist of the flex grid which is being used throughout the webpages as shown in the system. All of which is provided through the use of display attribute with the value flex which allow for more room for customization in terms of object orders, alignment and justifications, as well as one of the major implementation in flex objects which are the width size sizing themselves based on the minimum width of which fits the content. Additionally, the CSS present also include the use of media query which allows for more room for customization where the screen design changes based on the style applied in the media query. For instance, if the user's screen size fall under (min-width) or above (max-width) of the condition in the media query, the block of codes inside the media query will be executed and the styles will be applied to their respective elements.

## Homepage

```
/*Carousel */
.container-carousal {
  width: 80%;
  margin: 10px auto;
}
.carousel {
  border: 2px solid #ccc;
  height: 400px;
  position: relative;
  overflow: hidden;
}
.slider {
  height: 100%;
  display: flex;
  width: 400px;
  transition: all 0.3s;
}
.slider section {
  flex-basis: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
}
.slider img {
  max-width: 100%;
  height: auto;
  object-fit: cover;
}
.left, .right {
  position: absolute;
  transform: translateY(-50%);
  top: 50%;
  cursor: pointer;
}
.control .arrow i {
  font-size: 60px;
}
.arrow.left{
  left: 10px;
}
.arrow.right{
  right: 10px;
}
.control ul {
  position: absolute;
  bottom: 20px;
  left: 50%;
  transform: translate(-50%);
  list-style: none;
  display: flex;
  margin: 0;
  padding: 0;
}
.control ul li {
  width: 20px;
  height: 20px;
  border: 2px solid #333;
  border-radius: 50px;
  margin: 10px;
  background: #333;
  cursor: pointer;
}
.control ul li.selected {
  background: transparent;
}
/* END */

/* ===== */
Typography
=====
.heading-content p {
  border-top: 1px solid #ccc;
  padding-top: 0.5em;
  text-align: justify;
}
.service-container h2 {
  text-align: center;
  padding-bottom: 0.3em;
  border-bottom: 1px solid #ccc;
}
.highlight-gallery i,
.MCP-gallery i {
  color: #94C5CC;
  font-size: 2rem;
}
/* ===== */
Layout
=====
.container-flex {
  display: flex;
  flex-direction: column;
  width: 100%;
}
.heading-container {
  display: flex;
  width: 90%;
  margin: 0 auto;
  justify-content: center;
  border-bottom: 1px solid #ccc;
}
.heading-content {
  display: flex;
  flex-direction: column;
  margin: 0 auto;
  margin-top: 1em;
  width: 90%;
  padding: 0.4em;
  box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
}
.heading-content img {
  display: block;
  opacity: 0.9;
  width: 100%;
  max-width: 100%;
}
@media (min-width: 675px) {
  .heading-content {
    flex-direction: row;
  }
  .heading-content img {
    width: 50%;
    height: 15em;
  }
  .heading-content p {
    display: flex;
    align-items: center;
    padding-top: 0;
    border-top: none;
    padding-left: 0.5em;
    margin-left: 0.6em;
    border-left: 1px solid #ccc;
  }
}
```

A screenshot of a code editor window displaying a block of CSS code. The code is organized into several classes and media queries. It includes styling for containers, galleries, and specific elements like h2 and i. The CSS uses various properties such as width, margin, padding, font-size, and box-shadow. A media query at the bottom targets a minimum width of 675px and applies specific styles to the MCP-container and MCP-gallery classes.

```
.service-container { width: 90%; margin: 0 auto; margin-top: 1em; padding: 0.4em; padding-bottom: 1em; } .highlight-container { display: flex; flex-wrap: wrap; justify-content: space-evenly; } .highlight-gallery { text-align: center; padding-top: 0.5em; margin-top: 0.5em; width: 45%; box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2); } @media (min-width: 675px) { .highlight-gallery { width: 20%; margin-top: none; align-items: center; } .highlight-gallery i { font-size: 3.2rem; } } .medical-process-container { width: 90%; margin: 0 auto; padding: 0.4em; } .medical-process-container h2 { text-align: center; border-bottom: 1px solid #ccc; padding-bottom: 0.3em; } .MCP-container { display: flex; flex-wrap: wrap; justify-content: space-evenly; } .MCP-gallery { box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2); width: 45%; text-align: center; padding: 0.3em; } .MCP-last { width: 94%; margin-top: 1em; } @media (min-width: 675px) { .MCP-last { width: 30%; margin-top: 0; } .MCP-gallery { width: 30%; } .MCP-container { padding-bottom: 1em; } .MCP-gallery i { font-size: 3.2rem; } }
```

Figure 125 Homepage CSS

Based on the Figure 118 shown above, the source code as shown in the diagram shows how each component of homepage of Valley Clinic are being styled. In the CSS page for the respective page, multiple CSS components are used which include the class and id selector as well as html selectors are also being used to style the respective element in the HTML file. Aside from that, other attributes such as font-weight, paddings, text-alignment, margins, width, height, and etc. are some of the CSS attributes used in order to style the webpage. Additionally, one of the most significant piece of coding which consist of the flex grid which is being used throughout the

webpages as shown in the system. All of which is provided through the use of display attribute with the value flex which allow for more room for customization in terms of object orders, alignment and justifications, as well as one of the major implementation in flex objects which are the width size sizing themselves based on the minimum width of which fits the content. Additionally, the CSS present also include the use of media query which allows for more room for customization where the screen design changes based on the style applied in the media query. For instance, if the user's screen size fall under (min-width) or above (max-width) of the condition in the media query, the block of codes inside the media query will be executed and the styles will be applied to their respective elements. Aside from that, some elements also has transitions and effects added into their respective HTML elements such box-shadow being one of the most frequent used across multiple webpages to display galleries of information.

### **Medical Services CSS**

```

/*
=====
Typography
=====
.desc-col h3,
.consultation-fee h3,
.rating h3,
.doctor-col h3 {
    padding-bottom: 0.6em;
    border-bottom: 1px solid rgb(109, 107, 107);
}
.desc-col p {
    text-align:justify;
}
.doc-gallery h4 {
    color: #94C5CC;
}
/* .rating h3 {
    margin-bottom: 0.5em;
}
.checked {
    color: #B4D2E7;
} */

/*
=====
Layout
=====
main {
    width: 100%;
    padding: 0;
    background-color: #f8f8f8;
}
.container-flex {
    display: flex;
    flex-direction: column;
    width: 100%;
}
.heading-wrap {
    position: relative;
}
.heading-wrap::before {
    content: ' ';
    display: block;
    position: absolute;
    left: 0;
    top: 0;
    width: 100%;
    height: 100%;
    opacity: 0.6;
    background-image: url('../Images/bg-img-TS.jpg');
    background-repeat: no-repeat;
    background-position: 50% 0;
    background-size: cover;
}
.medical-heading {
    position: relative;
    display: flex;
    justify-content: center;
    height: 10em;
    align-items: center;
}

```

```

.desc-col,
.misc-col,
.doctor-col {
    width: 90%;
    margin: 0 auto;
    margin-top: 1em;
    padding: 0.5em;
    border: 2px solid #94C5CC;
    border-radius: 10px;
    box-shadow: 5px 4px 26px 0px #6D6B6B;
}
.doctor-col {
    display: flex;
    flex-direction: column;
}
.doc-gallery {
    display: flex;
    flex-direction: column;
    width: 100%;
    justify-content: space-evenly;
    align-items: center;
}
.doctor-col img {
    display: block;
    width: 35%;
    border: 1px solid rgb(109, 107, 107);
    border-radius: 50%;
    max-width: 100%;
}
@media (min-width: 675px) {
    .doc-gallery-container {
        display: flex;
        flex-direction: row;
        width: 100%;
    }
    .desc-col,
    .misc-col {
        margin: 0;
        width: 40%;
        margin: 1.5em;
    }
    .doctor-col {
        width: 90%;
    }
    .body-content {
        display: flex;
        flex-direction: row;
        flex-wrap: wrap;
        justify-content: space-evenly;
    }
    .doctor-col img {
        width: 22%;
    }
}

```

Figure 126 Medical Services CSS

Based on the Figure 119 shown above, the source code as shown in the diagram shows how each component of medical services webpage are being styled. In the CSS page for the respective page, multiple CSS components are used which include the class and id selector as well as html selectors are also being used to style the respective element in the HTML file. Aside from that, other attributes such as font-weight, paddings, text-alignment, margins, width, height, and

etc. are some of the CSS attributes used in order to style the webpage. Additionally, one of the most significant piece of coding which consist of the flex grid which is being used throughout the webpages as shown in the system. All of which is provided through the use of display attribute with the value flex which allow for more room for customization in terms of object orders, alignment and justifications, as well as one of the major implementation in flex objects which are the width size sizing themselves based on the minimum width of which fits the content. Additionally, the CSS present also include the use of media query which allows for more room for customization where the screen design changes based on the style applied in the media query. For instance, if the user's screen size fall under (min-width) or above (max-width) of the condition in the media query, the block of codes inside the media query will be executed and the styles will be applied to their respective elements.

## **Heading and Footer**

```

/*
=====
Layout
=====
*/
.container {
    width: 100%;
    margin: 0 auto;
    max-height: 120px;
}
.container-nav {
    margin-left: 0;
    display: flex;
    justify-content: space-between;
}
header {
    background-color: #f8f8f8;
}
@media (max-width: 675px) {
    .container {
        max-height: none;
    }
    .container-nav {
        flex-direction: column;
    }
    .header {
        text-align: center;
    }
}
header img {
    width: 100%;
    max-width: 170px;
}
.header-title {
    margin: 1em;
    display: flex;
    justify-content:center;
}
@media (max-width: 675px) {
    .header-title {
        margin-bottom: 0;
        text-align: center;
    }
    header img {
        width: 50%;
    }
}
/* Footer */
.footer {
    display: flex;
    margin-top: 2.5em;
    flex-direction: column;
    padding: 0.5em;
    background-color: #A1A6B4;
}
.footer-box {
    display: flex;
    flex-direction: row;
}
.follow-us {
    display: flex;
    flex-direction: column;
}
.follow-box {
    display: flex;
    justify-content: left;
    margin: 0.5em;
}
@media (min-width: 675px) {
    div.footer {
        display: flex;
        flex-direction: row;
        justify-content: space-between;
    }
    .about-us {
        width: 30%;
    }
    .contact-us {
        width: 37%;
    }
    .follow-us {
        width: 30%;
    }
}

```

Figure 127 Heading and Footer CSS

Based on the Figure 120 shown above, the source code as shown in the diagram shows how each component of heading and footer are being styled. With the help of the element selectors such as the class and id selector as well as HTML selectors, each elements of the header and footer are styled respectively though the use of various CSS attributes of values. For instance, the some fonts which consist of hyperlinks to the Clinic's socials are being styled with their respective colors

such as the link to Valley Clinic's Facebook Page is set to blue while Valley Clinic's Instagram link is set to orange. Aside from that, the about us section, contact details, as well as the social links are set around a flex container which holds these component together with their respective sizes. Additionally, media queries are also being used in order to style the HTML page accordingly if cases where the screen width decreases due to different platforms being used.

## Our doctor CSS

```

/* =====
Typography
===== */
.title-container h2 {
    color: #94C5CC;
    text-align: center;
    border-bottom: 1px solid #EEEEEE;
}
.title-container p {
    text-align: justify;
}
.doctor-gallery h3{
    color: #000100;
    font-size: 1rem;
    text-align: center;
}
.doctor-gallery h4 {
    color: #A1A6B4;
    text-align: center;
}
/*=====
Layout
=====*/
.container-flex,
.header-content {
    display: flex;
    width: 100%;
}

.header-content img {
    width: 100%;
    opacity: 0.5;
    height: 15em;
}
.title-container {
    width: 90%;
    margin: 0 auto;
    margin-top: 0.5em;
}
@media (max-width: 675px) {
    .title-container h2 {
        font-size: 1.2rem;
    }
    .title-container p {
        font-size: 0.83rem;
    }
}
.body-container {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-evenly;
    width: 90%;
    margin: 0 auto;
    padding: 1em;
    /*Border - wrap content
    border: 1px solid #A1A6B4;
    border-radius: 10px; */
}
.doctor-gallery img {
    display: block;
    max-width: 60%;
    width: 100%;
    margin: 0 auto;
}
.doctor-gallery {
    display: flex;
    flex-direction: column;
    width: 40%;
    margin-top: 1em;
    box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
}
@media (min-width: 675px) {
    .doctor-gallery {
        width: 30%;
    }
}

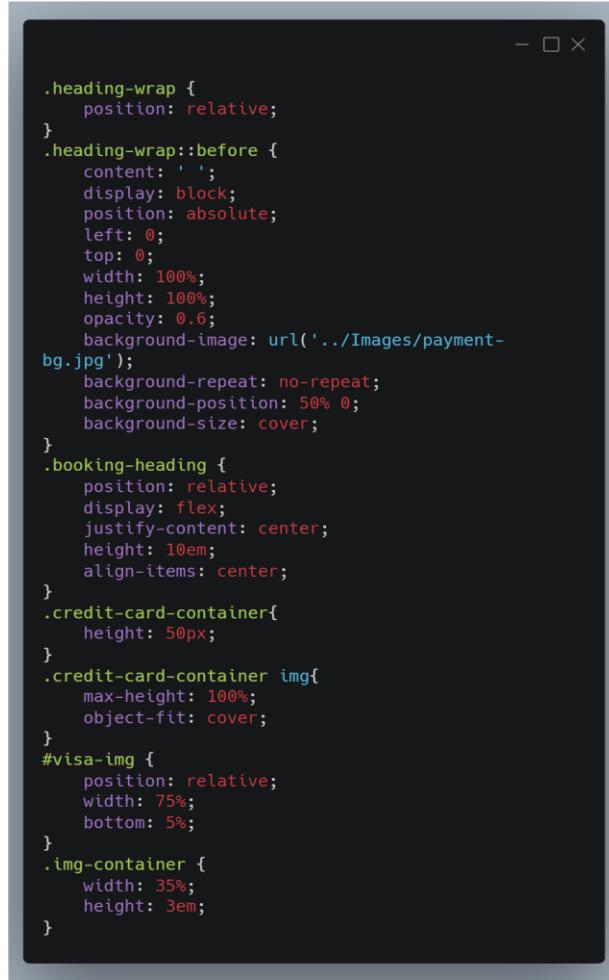
```

Figure 128 Our Doctor page CSS

Based on the Figure 121 shown above, the source code as shown in the diagram shows how each component doctor details page are being styled. For instance, a set of CSS attributes are being used to style the gallery of which the doctor picture is shown as well as styling the heading which consist of the doctor's name and the field, they specialized in. Aside from that, the container with the flex as the display is also set to dynamically hold the entire gallery within the webpage.

.In the CSS page for the respective page, multiple CSS components are used which include the class and id selector as well as html selectors are also being used to style the respective element in the HTML file. Aside from that, other attributes such as font-weight, paddings, text-alignment, margins, width, height, and etc. are some of the CSS attributes used in order to style the webpage. Additionally, one of the most significant piece of coding which consist of the flex grid which is being used throughout the webpages as shown in the system. All of which is provided through the use of display attribute with the value flex which allow for more room for customization in terms of object orders, alignment and justifications, as well as one of the major implementation in flex objects which are the width size sizing themselves based on the minimum width of which fits the content. Additionally, the CSS present also include the use of media query which allows for more room for customization where the screen design changes based on the style applied in the media query. For instance, if the user's screen size fall under (min-width) or above (max-width) of the condition in the media query, the block of codes inside the media query will be executed and the styles will be applied to their respective elements.

## **Payment CSS**

A screenshot of a code editor window displaying CSS code. The code is organized into several classes and a pseudo-element selector. It includes styling for a background image, flexbox layout, and specific dimensions for elements like credit card containers and Visa logos.

```
.heading-wrap { position: relative; } .heading-wrap::before { content: ' '; display: block; position: absolute; left: 0; top: 0; width: 100%; height: 100%; opacity: 0.6; background-image: url('../Images/payment-bg.jpg'); background-repeat: no-repeat; background-position: 50% 0; background-size: cover; } .booking-heading { position: relative; display: flex; justify-content: center; height: 10em; align-items: center; } .credit-card-container { height: 50px; } .credit-card-container img { max-height: 100%; object-fit: cover; } #visa-img { position: relative; width: 75%; bottom: 5%; } .img-container { width: 35%; height: 3em; }
```

Figure 129 Payment CSS

Based on Figure 122 which shows the Payment CSS source code used for styling the specified elements in the webpage. The CSS which involves the use of various HTML selectors as well as class and id selectors are used to style individual elements in the webpage. Aside from that, the CSS also styles the size in terms of height and width of the pictures and the container holding the picture resides in such as the credit card images.

## Treatment and Services CSS

```
/* =====
Typographpy
===== */
.header-img img{
    display: block;
    max-width: 100%;
    max-height: 300px;
    width: 100%;
    opacity: 0.5;
}
.ts_gallery a,
.treatment-highlight h3 {
    font-size: 0.9375rem;
}
.ts_gallery a {
    font-weight: 600;
    text-decoration: none;
    color: black;
}
.ts_gallery a:hover {
    color: grey;
}
.treatment-highlight h3 {
    margin: 0;
}
.title-content2 h2 {
    font-size: 1.2rem;
}
.equipment-content h4 {
    padding-bottom: 0.5em;
    border-bottom: 1px solid #eee;
}
.service-overview p {
    text-align: justify;
}
```

```
li.nav-next {
    font-size: 1rem;
    border-radius: 5px 25px 25px 5px;
}
li.nav_prev {
    font-size: 1rem;
    border-radius: 25px 5px 5px 25px;
}
li.nav-numb:hover,
li.nav-prev:hover,
li.nav-next:hover {
    background: #A1A6B4;
}

.pagination-menu a {
    text-decoration: none;
}

@media (min-width: 675px) {
    .ts_gallery {
        padding-bottom: 0.5em;
    }
    .content2-body {
        text-align: left;
        flex-direction: row;
        align-items: center;
    }
    .content2-img img {
        width: 350em;
        max-width: 100%;
        margin: 0;
        border-right: 1px solid #A1A6B4;
    }
    .equipment-content img{
        max-width: 22%;
        margin: 0.5em;
        border: 1px solid #A1A6B4;
    }
    .service-overview {
        display: flex;
        flex-direction: column;
        margin-left: 1em;
        justify-content: center;
    }
}
```

```

* =====
Layout
=====
.container-flex {
  width: 100%;
  display: flex;
  border-bottom: none;
}
main {
  width: 100%;
}
.search-panel {
  justify-content: center;
  text-align: center;
}
.search-panel h2 {
  text-decoration: underline;
}
.search-bar input[type="text"] {
  font-size: 0.95rem;
  font-family: inherit;
  text-align: left;
  padding: 0.3em;
  width: 30%;
  border-radius: 4px;
  box-sizing: border-box;
  border: 2px solid #f0f0f0;
  border-radius: 4px;
}
.search-bar input:focus {
  outline: 0;
  border-color: #777;
}
.search-bar button {
  font-size: 1.25rem;
}
.content-body {
  display: flex;
  flex-direction: column;
  margin: 0 auto;
  margin-top: 1em;
  width: 90%;
  max-width: 100%;
}
.main-content {
  display: flex;
  order: 1;
  flex-wrap: wrap;
  justify-content: space-between;
  padding: 1em;
  width: 90%;
  margin: 0 auto;
  border: 1px solid #A1A6B4;
}
.ts_gallery {
  margin: 0.5em;
  text-align: center;
  width: 20%;
  min-width: 180px;
  border: 1px solid black;
  border-radius: 10px;
  padding: 0;
  overflow: hidden;
}
.ts_gallery img {
  width: 100%;
  display: block;
  max-width: 100%;
}
.content-subtitle {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-evenly;
  width: 90%;
  margin: 0 auto;
  margin-top: 1em;
  padding: 1em;
}
.treatment-highlight {
  text-align: center;
  width: fit-content;
  max-width: 20%;
  margin: 0.5em;
  padding: 1em;
  border: 1px solid #A1A6B4;
  border-radius: 15px;
}
.title-content2 {
  display: block;
  width: 90%;
  margin: 0 auto;
  border-bottom: 1px solid #A1A6B4;
}
.content2-body {
  display: flex;
  flex-direction: column;
  text-align: center;
  width: 90%;
  margin: 0 auto;
  margin-top: 1em;
  padding: 0.5em;
  box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
}
.content2-body img {
  display: block;
  max-width: 100%;
}
.content2-img img {
  width: 100%;
  max-width: 600px;
  margin: 0 auto;
}
.equipment-content img{
  max-width: 45%;
  border-radius: 50%;
  margin: 0.5em;
  border: 1px solid #A1A6B4;
}
.equipment-gallery {
  display: flex;
  flex-wrap: wrap;
}
/* Pagination */
.pagination {
  text-align: center;
  width: 100%;
  margin: 0 auto;
  order: 2;
  margin-top: 1em;
}
.pagination ul {
  list-style-type: none;
  overflow: hidden;
  width: fit-content;
  margin: 0 auto;
  background-color: #94C5CC;
  border-radius: 50px;
}
.pagination-menu li{
  display: inline-block;
  padding: 0.5em;
  width: 1.5em;
  max-width: 100%;
  cursor: pointer;
  transition: all 0.3s ease;
}
.pagination-menu li a {
  color: black;
}
li.nav-numb {
  font-size: 1rem;
  border-radius: 50%;
}

```

Figure 130 Treatment and Services CSS

Based on the Figure 123 shown above, the source code as shown in the diagram shows how each component of treatment and services webpage are being styled. In the CSS page for the

respective page, multiple CSS components are used which include the class and id selector as well as html selectors are also being used to style the respective element in the HTML file. Aside from that, other attributes such as font-weight, paddings, text-alignment, margins, width, height, and etc. are some of the CSS attributes used in order to style the webpage. Additionally, one of the most significant piece of coding which consist of the flex grid which is being used throughout the webpages as shown in the system. All of which is provided through the use of display attribute with the value flex which allow for more room for customization in terms of object orders, alignment and justifications, as well as one of the major implementation in flex objects which are the width size sizing themselves based on the minimum width of which fits the content. Additionally, the CSS present also include the use of media query which allows for more room for customization where the screen design changes based on the style applied in the media query. For instance, if the user's screen size fall under (min-width) or above (max-width) of the condition in the media query, the block of codes inside the media query will be executed and the styles will be applied to their respective elements.

## **JavaScript**

### **Booking JavaScript**

```

//Function to display available slot form
const nextBtn = document.querySelector('.next-btn');
const fieldset = document.querySelector('.fieldset');
const hideSection = document.querySelector('.control-hidden');
const checkBtn = document.getElementById('checkBtn');

nextBtn.addEventListener('click', (e) => {
  e.preventDefault();
  checkBtn.click();
  fieldset.style.opacity= "0.5";
  hideSection.removeAttribute("hidden");
});

const finalBtn = document.querySelector(".submitBtn")
const form1 = document.querySelector(".form1")

finalBtn.addEventListener('click', (d) => {
  fieldset.removeAttribute("disabled");
})

// Declare Initial Datetime
let startDatetime = new Date()
startDatetime.setHours(8);
startDatetime.setMinutes(0);
startDatetime.setSeconds(0);
console.log(startDatetime);

//Loop for slots
const slotsArray = [];
for( let i = 0; i < 22; i++ ) {
  //Add Time Iteration
  let startMin = startDatetime.getMinutes();
  startDatetime.setMinutes(startMin + 30);
  // console.log('Slots: ${startDatetime}');

  //Append slots into an Array
  let slotHrs = startDatetime.getHours();
  let slotMins = startDatetime.getMinutes();
  let stringMins = slotMins.toString();
  let stringHrs = slotHrs.toString();

  let slotPeriods = `${slotHrs}.${slotMins}`;
  slotsArray.push(slotPeriods);
}

//Array
arrayLength = slotsArray.length;
//START
checkBtn.addEventListener("click", postDate);

function postDate(e){
  e.preventDefault();
  var appt_slot = document.getElementById("appt_date").value;
  var param = "appt_date=" + appt_slot;

  fieldset.setAttribute("disabled", "disabled"); //Disable first form

  var xhr = new XMLHttpRequest();
  xhr.open('POST', 'check_slots.php', true);
  xhr.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
  xhr.onload = function(){
    if(this.status == 200){
      let takenSlotsArray = JSON.parse(xhr.responseText);
      const tempArray = [];

      for (let i = 0; i < takenSlotsArray.length; i++) {
        tempArray.push(takenSlotsArray[i].slice(0,5).replace(":", "."));
      }

      for(let i = 0; i < arrayLength; i++) {
        const slots = document.createElement("button");
        const bookingContainer = document.querySelector('.booking_slots');
        bookingContainer.appendChild(slots);
        slots.innerHTML = slotsArray[i];
        slots.setAttribute("value", slotsArray[i]);
        for ( let j = 0; j < tempArray.length; j++ ) {
          if (tempArray[j] == slotsArray[i]) {
            slots.setAttribute("disabled", "disabled");
          }
        }
      }

      const appendSlot = document.querySelectorAll('.booking_slots button')
      appendSlot.forEach(function(btn) {
        btn.setAttribute("type", "submit");
        btn.setAttribute("class", "btn btn-outline-info m-1 btn-sm");
        btn.setAttribute("name", "chosen_slot");
      });
    }
  }
  //Patient chosen time slot
  appendSlot.forEach(function(btn) {
    btn.addEventListener("click", checkValue);
  })
  function checkValue() {
    const temp_var = document.getElementById('patient_chosen_slot');
    let slotValues = this.value;
    temp_var.value = slotValues;
  }
  let takenPeriods = document.getElementById('arrayRepository')
  takenPeriods.value = tempArray;
}

xhr.send(param);
}

```

Figure 131 JavaScript Bookings

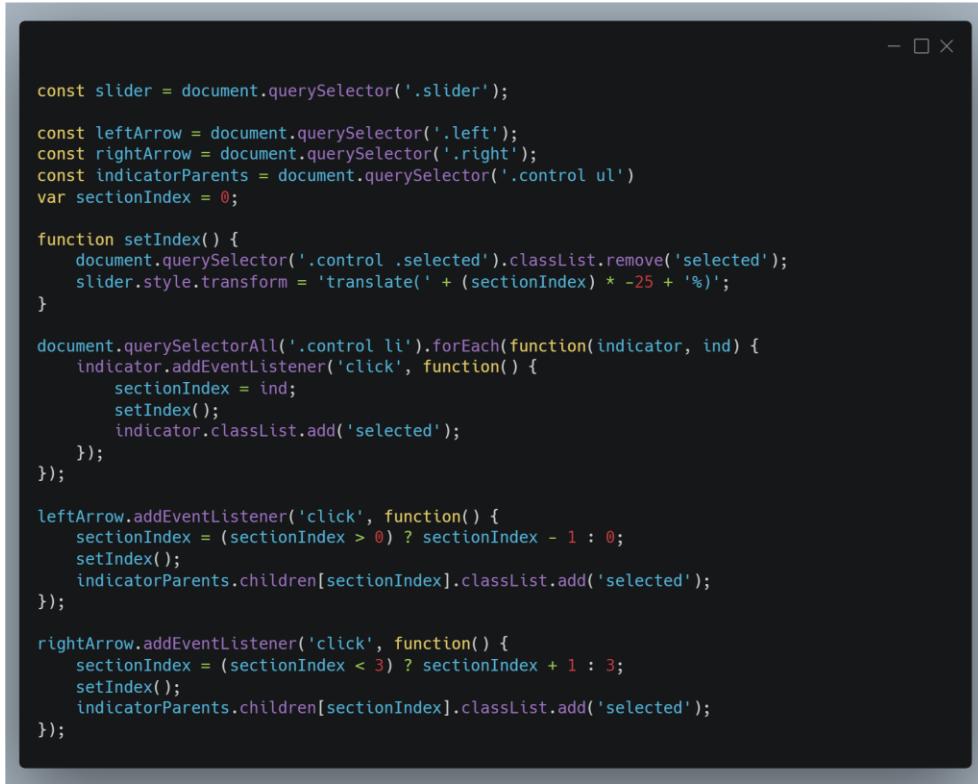
JavaScript is a programming language which consist of one of the core component or the building block of a website. JavaScript are commonly used in web development although might prove to be useful in some other environments as well. Therefore, various JavaScript codes can be seen being used throughout the Clinic Management System. In this case, JavaScript is being used in booking webpage which perform various functionalities such as for the use of asynchronous communication between the server and PHP or HTML files. First and foremost, one of the most significant piece of coding which would be to display the iterated booking slots which are available for the chosen date. By using JavaScript syntax and the combination with various operators, a JavaScript statement are formed. For instance, various variables are being assigned in the global environment and with the combination of query selector function, more actions will be able to be perform on that particular fetched data. This can be seen being used in the booking section where the user clicks on the next button which is equip with an event listener, the function preventDefault() is then executed to stop the automated submission upon clicking the submit button and the rest of the code which focuses on styling the available slots container by removing the hidden attribute from the HTML element. Aside from that, the buttons which is used to select the slots provided are appended via the setAttribute() function. First and foremost, the logic behind the operation of distribution of slots starts with the declaration of the variable startDatetime and new Date() function. The hours, minutes and seconds are then assigned into the declared variable. Next, an array is declared which will be used to store all the iterated slots. A for loop is used and inside the loop, the declared starting slots are iterated by adding 30 minutes and finally being pushed into the array via push() function. A conditional statement is also located in the loop in order to add missing zeros into the hours and minutes of the time slot for display purposes.

Another event listener is then added into next button which prevent the form from submitting the form prior before executing the entire source code, a function postdate() declared to be used as to perform asynchronous function through XMLHttpRequest. An object of XMLHttpRequest is store in a variable named xhr which will be used to submit the chosen date of appointment to check\_slots.php which will check for any slots booked by the user. Upon success of execution, the JSON object are then stored into a tempArray array which consist of the taken slots. Another loop will then be used to iterate through the time slots and the button which can be seen by the user is created through the createElement() function with the value of the iterated time slots from the first loop mentioned. While still in the same loop, another loop is started to loop

.

through the booked array which was previously obtained from check\_slots.php. If the slots value matches the taken slots, then the slot will be set with the attribute disabled. Next, the button is then styled by adding attributes with the setAttribute function by addition the type button, class for bootstrap styling, and name of the button which will be used in the next associated PHP file to acquire the value assigned to the button.

## Carousel



A screenshot of a code editor window displaying a block of JavaScript code. The code is written in a light blue font on a dark background. It defines variables for a slider, left and right arrows, and indicator parents. It includes a function to set the index of the slider and another to handle click events on the indicators. The code uses document.querySelectorAll and .classList.add to manage the state of the elements.

```
const slider = document.querySelector('.slider');

const leftArrow = document.querySelector('.left');
const rightArrow = document.querySelector('.right');
const indicatorParents = document.querySelector('.control ul')
var sectionIndex = 0;

function setIndex() {
    document.querySelector('.control .selected').classList.remove('selected');
    slider.style.transform = 'translate(' + (sectionIndex * -25 + '%')';
}

document.querySelectorAll('.control li').forEach(indicator, ind) {
    indicator.addEventListener('click', function() {
        sectionIndex = ind;
        setIndex();
        indicator.classList.add('selected');
    });
};

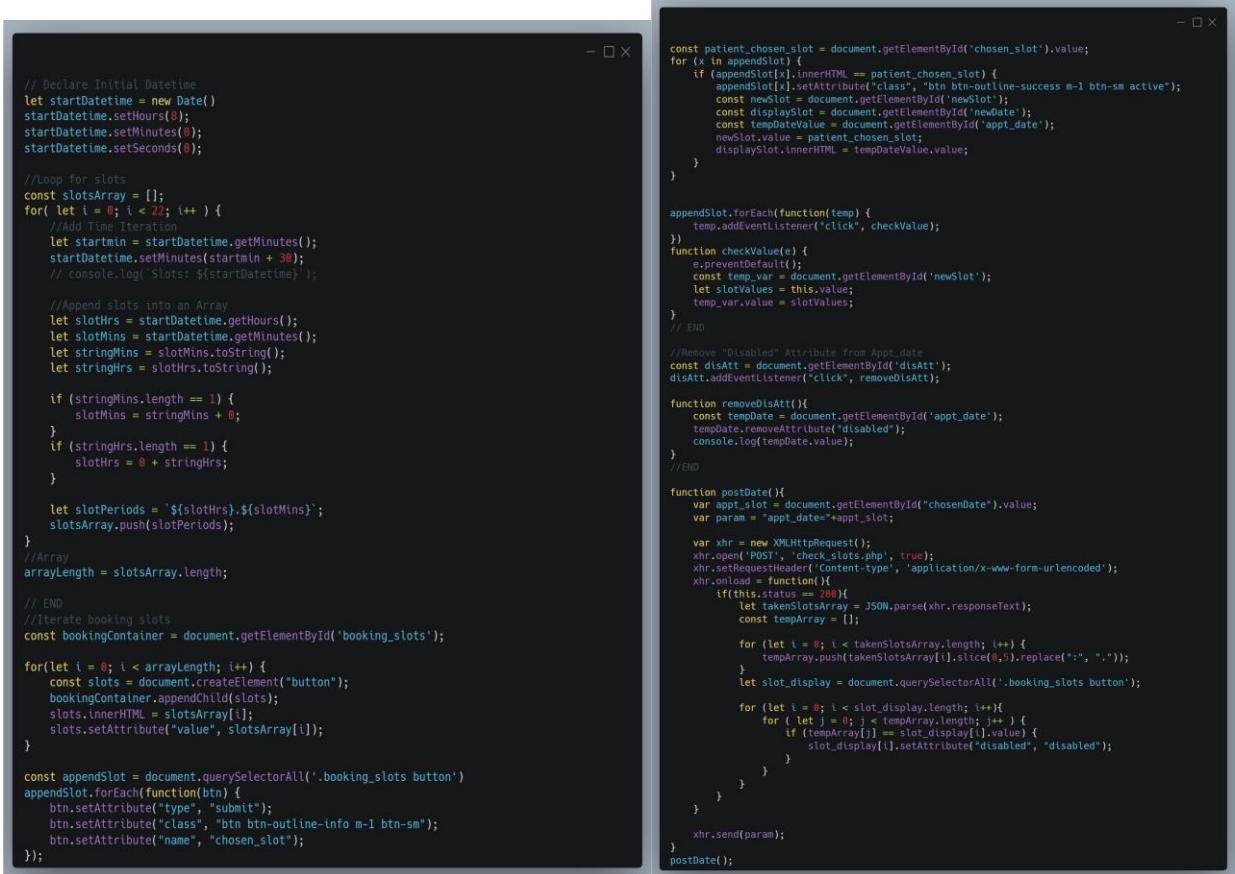
leftArrow.addEventListener('click', function() {
    sectionIndex = (sectionIndex > 0) ? sectionIndex - 1 : 0;
    setIndex();
    indicatorParents.children[sectionIndex].classList.add('selected');
});

rightArrow.addEventListener('click', function() {
    sectionIndex = (sectionIndex < 3) ? sectionIndex + 1 : 3;
    setIndex();
    indicatorParents.children[sectionIndex].classList.add('selected');
});
```

Figure 132 Carousel JavaScript

Similarly, to the booking source code, the carousel which is located in the homepage of Valley Clinic Management System allows user's interaction by using JavaScript event listeners and query selector function.

## Confirm Booking



The screenshot shows a browser window with two code editors side-by-side. The left editor contains the main logic for creating and validating booking slots, while the right editor contains the logic for handling the selected slot and sending a POST request to a PHP file.

```
// Declare Initial Datetime
let startDatetime = new Date();
startDatetime.setHours(8);
startDatetime.setMinutes(0);
startDatetime.setSeconds(0);

//Loop for slots
const slotsArray = [];
for( let i = 0; i < 22; i++ ) {
    //Add Time Iteration
    let startMin = startDatetime.getMinutes();
    startDatetime.setMinutes(startMin + 30);
    // console.log('Slots: ${startDatetime}');

    //Append slots into an Array
    let slotHrs = startDatetime.getHours();
    let slotMins = startDatetime.getMinutes();
    let stringMins = slotMins.toString();
    let stringHrs = slotHrs.toString();

    if (stringMins.length == 1) {
        slotMins = stringMins + 0;
    }
    if (stringHrs.length == 1) {
        slotHrs = 0 + stringHrs;
    }

    let slotPeriods = `${slotHrs}.${slotMins}`;
    slotsArray.push(slotPeriods);
}
//Array
arrayLength = slotsArray.length;

// END
//Iterate booking slots
const bookingContainer = document.getElementById('booking_slots');

for(let i = 0; i < arrayLength; i++) {
    const slots = document.createElement("button");
    bookingContainer.appendChild(slots);
    slots.innerHTML = slotsArray[i];
    slots.setAttribute("value", slotsArray[i]);
}

const appendSlot = document.querySelectorAll('.booking_slots button')
appendSlot.forEach(function(btn) {
    btn.setAttribute("type", "submit");
    btn.setAttribute("class", "btn btn-outline-info m-1 btn-sm");
    btn.setAttribute("name", "chosen_slot");
});
```

```
const patient_chosen_slot = document.getElementById('chosen_slot').value;
for (x in appendSlot) {
    if (appendSlot[x].innerHTML == patient_chosen_slot) {
        appendSlot[x].setAttribute("class", "btn btn-outline-success m-1 btn-sm active");
        const newSlot = document.getElementById('newSlot');
        const displaySlot = document.getElementById('newDate');
        const tempDateValue = document.getElementById('appt_date');
        newSlot.value = patient_chosen_slot;
        displaySlot.innerHTML = tempDateValue.value;
    }
}

appendSlot.forEach(function(temp) {
    temp.addEventListener("click", checkValue);
})
function checkValue() {
    e.preventDefault();
    const temp_var = document.getElementById('newSlot');
    let slotValues = this.value;
    temp_var.value = slotValues;
}
// END

//Remove "Disabled" Attribute from Appt_date
const disAtt = document.getElementById('disAtt');
disAtt.addEventListener("click", removeDisAtt);

function removeDisAtt(){
    const tempDate = document.getElementById('appt_date');
    tempDate.removeAttribute("disabled");
    console.log(tempDate.value);
}
//END

function postData(){
    var app_slot = document.getElementById("chosenDate").value;
    var param = `appt_date=${app_slot}`;

    var xhr = new XMLHttpRequest();
    xhr.open('POST', 'check_slots.php', true);
    xhr.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
    xhr.onload = function(){
        if(this.status == 200){
            let takenSlotsArray = JSON.parse(xhr.responseText);
            const tempArray = [];

            for (let i = 0; i < takenSlotsArray.length; i++) {
                tempArray.push(takenSlotsArray[i].slice(0,5).replace(":","."));
            }

            let slot_display = document.querySelectorAll('.booking_slots button');

            for (let i = 0; i < slot_display.length; i++) {
                for ( let j = 0; j < tempArray.length; j++ ) {
                    if (tempArray[j] == slot_display[i].value) {
                        slot_display[i].setAttribute("disabled", "disabled");
                    }
                }
            }

            xhr.send(param);
        }
        postData();
    }
}
```

Figure 133 Confirm Booking Details JavaScript

Similar to the JavaScript described in booking.js, the main feature of which provided by payment.js would be for iterating through the available slots and perform validation checking on the availability of the chosen slots before assigning the value of which chosen by the user into an input tag before being submitted into the associated PHP file.

```
function scrollToCash (){
    const toDiv = document.getElementById('cash-tab');
    const remvInvis = document.getElementById('invisTab');
    // const ccVar = document.getElementById('credit_card-tab')
    // ccVar.setAttribute("class", "nav-link disabled");

    remvInvis.setAttribute("class", "container m-3 mx-auto visible")
    toDiv.scrollIntoView();
}

function scrollToCC(){
    const toDiv = document.getElementById('credit_card-tab');
    const remvInvis = document.getElementById('invisTab');

    toDiv.click(); //Set event to trigger tab

    remvInvis.setAttribute("class", "container m-3 mx-auto visible");
    toDiv.scrollIntoView();
}
```

Figure 134 Payment JavaScript

In payment.js list two function which will be called upon selection of payment method. From the first defined function, the tab of which located the cash payment gateway is set to be visible through the use of `setAttribute` function and a `scrollIntoView` is also used to scroll the user's screen to the defined location. Similarly, the second function also performs the same feature but in this case for credit card payment gateway. All of which allow the user to make payment based on their selected payment method while scrolling into the specified element into the visible area of the browser.

## 18.7 Administrator Module

### 18.7.1 Create – HTML Form, PHP

#### Create Customer Account Table

```
CREATE TABLE IF NOT EXISTS `tbl_account` (
  `Acc_ID` int NOT NULL AUTO_INCREMENT,
  `Name` varchar(50) DEFAULT NULL,
  `Phone_number` varchar(50) DEFAULT NULL,
  `Email` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`Acc_ID`)
)
```

Figure 135: Create Account Table

This is the SQL query. The CREATE TABLE statement creates a new table in the database "valleyclinic". Each column is followed by a datatype parameter that determines the type of data that may be inserted in that column. The VARCHAR datatype accepts characters and integers up to a certain size. When a new row is added, AUTO INCREMENT will raise the number of Acc ID. The PRIMARY KEY enables the database to recognize each account individually.

## Create Doctor Table

```
CREATE TABLE `tbl_doctor` (
    `DocID` int NOT NULL AUTO_INCREMENT,
    `Name` varchar(50) NOT NULL,
    `Contact_number` varchar(50) NOT NULL,
    `Address` text NOT NULL,
    `Email` varchar(255) NOT NULL,
    `Specialist_category` varchar(255) NOT NULL,
    `Gender` varchar(10) NOT NULL,
    `DOB` date NOT NULL,
    `Profile_pic` longblob NOT NULL,
    `Username` varchar(20) NOT NULL,
    `Password` varchar(20) NOT NULL,
    PRIMARY KEY (`DocID`)
)
```

Figure 136: Create Doctor Table

In the "valleyclinic" database, the SQL CREATE TABLE statement below creates a new table. Each column is followed by a datatype parameter that determines the type of data that may be inserted in that column. VARCHAR, DATE, INT, TEXT, and LONGLOB were the datatypes used to create the book table. The VARCHAR datatype takes characters and integers up to a certain size, but the INT datatype only accepts whole numbers. DATE accepts date-formatted data. LONGLOB accepts binary big object and TEXT accepts string value. Finally, a PRIMARY KEY is defined to enable the database to identify each doctor in the database individually.

## Create New Database Connection

```
<?php
$conn = mysqli_connect("localhost", "root", "", "valleyclinic", "3306");
// Check connection
if (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
?>
```

Figure 137: conn.php

In the connection file, `mysqli_connect()` function is used to create a new MySQL server connection. The parameters included in this function are host, username, password, database name and port number. If the connection has error, there will be an error message showing the error.

## Create Session

```
<?php

if(!isset($_SESSION))
{
    session_start();
}

$admin_username = $_SESSION['Admin_Username'];

if (!isset($_SESSION['Admin_Username']))
{
    header("location: admin_login.php");
}

?>
```

Figure 138: admin\_session.php

In the session file, if there is no session variable which contains associative array, session start () function will create or continue the current session. Admin\_username session is used to store variable of admin\_username. If there is no Admin\_Username session, admin will be redirected to the admin login page.

## Create New Treatment and Services

```
<a href="create_services.php">
<button type="button" class="btn btn-dark">New Treatment & Services <i class="bi bi-plus-circle-fill ms-1"></i></button>
</a>
```

Figure 139: New Treatment and Services Button

This button is placed top right of the treatment and services page. Once admin click on this new Treatment and services button, admin will be led to the create services page using the anchor tag.

```
<form class="mx-5 vh-100" action="insert_services.php" method="POST" ENCTYPE="multipart/form-data">
  <div class="mb-3">
    <label for="Title" class="form-label">Title</label>
    <input class="form-control" type="text" placeholder="Title" name="Title" aria-label="Title" required>
  </div>
  <div class="mb-3">
    <label for="Description" class="form-label">Description</label>
    <textarea class="form-control" placeholder="Description" style="height: 100px" name="Description" aria-label="Description" required></textarea>
  </div>
  <div class="mb-3">
    <label for="Specialist_category" class="form-label">Specialist Category</label>
    <input class="form-control" type="text" placeholder="Specialist Category" name="Specialist_category" aria-label="Specialist Category" required>
  </div>
  <div class="mb-3">
    <label for="Consultation_Fees" class="form-label">Consultation Fee Range</label>
    <select class="form-select" aria-label="Consultation Fee" name="Consultation_Fees" required>
      <option selected>Please Select A Range</option>
      <option value="RM30 - RM5">RM30 - RM50</option>
      <option value="RM51 - RM80">RM51 - RM80</option>
      <option value="RM81 - RM125">RM81 - RM125</option>
      <option value="RM126 - RM235">RM126 - RM235</option>
    </select>
  </div>
  <div class="mb-3">
    <label for="Picture" class="form-label">Picture</label>
    <div class="input-group mb-3">
      <input type="file" class="form-control" name="Picture" required>
    </div>
  </div>
  <div class="d-grid">
    <button type="submit" class="btn btn-dark mt-5">Submit</button>
  </div>
</form>
```

Figure 140: Create New Treatment and Services Form

This is the HTML form of create new treatment and services. This form contains input field for the title, description, specialist category, consultation fee range and picture of the treatment. This form is using POST method and the code in the insert\_services.php will be executed once the

submit button is clicked. The enctype attribute value is set as “multipart/form-data” to upload file through the form.

```
$check_existing = mysqli_query($conn,"SELECT * FROM `tbl_medical_services` WHERE Title = '".$_POST[Title]'");
$exist_rows = mysqli_num_rows($check_existing);

if (!$exist_rows == 0)
{
    header("Location: create_services.php?error= Title already exists! Please check again.");
}
else
{
    $target_dir = "treatment_images/";
    $target_file = $target_dir . basename($_FILES["Picture"]["name"]);
    if (move_uploaded_file($_FILES["Picture"]["tmp_name"], $target_file)){
        //To get file name
        $file_name = basename($_FILES["Picture"]["name"]);
        $sql="INSERT INTO `tbl_medical_services` (Title, Description, Specialist_category, ConsultationFees, Picture)
        VALUES
        ('$_POST[Title]','$_POST[Description]','$_POST[Specialist_category]','$_POST[Consultation_Fees]',
        '$file_name')";

        if (!mysqli_query($conn,$sql)){
            die('Error: ' . mysqli_error($conn));
        }
        else {
            header("Location: create_services.php?success= New Treatment & Services Successfully Created !");
        }
    } else {
        header(
        "Location: create_services.php?error= Sorry, there was an error uploading your file. Please check again.");
    }
}
```

Figure 141: insert\_services.php

This new treatment and services function enables admin to insert new treatment and services into the treatment and services database. With the POST method, the treatment and services information will be accepted by the new treatment and services form. Once the form is submitted, the treatment and services information will be retrieved using the PHP POST method. The SQL INSERT INTO statement is used to define column names and values to be inserted into the database to insert the book. The query will be executed by the PHP function mysqli\_query(). Finally, after the treatment and services has been uploaded to the database, the warning message or success message will be shown above the form.

## Create New Doctor Account



Figure 142: New Account Button

This button is in the search doctor page where admin can navigate to the create doctor page by clicking on this button.

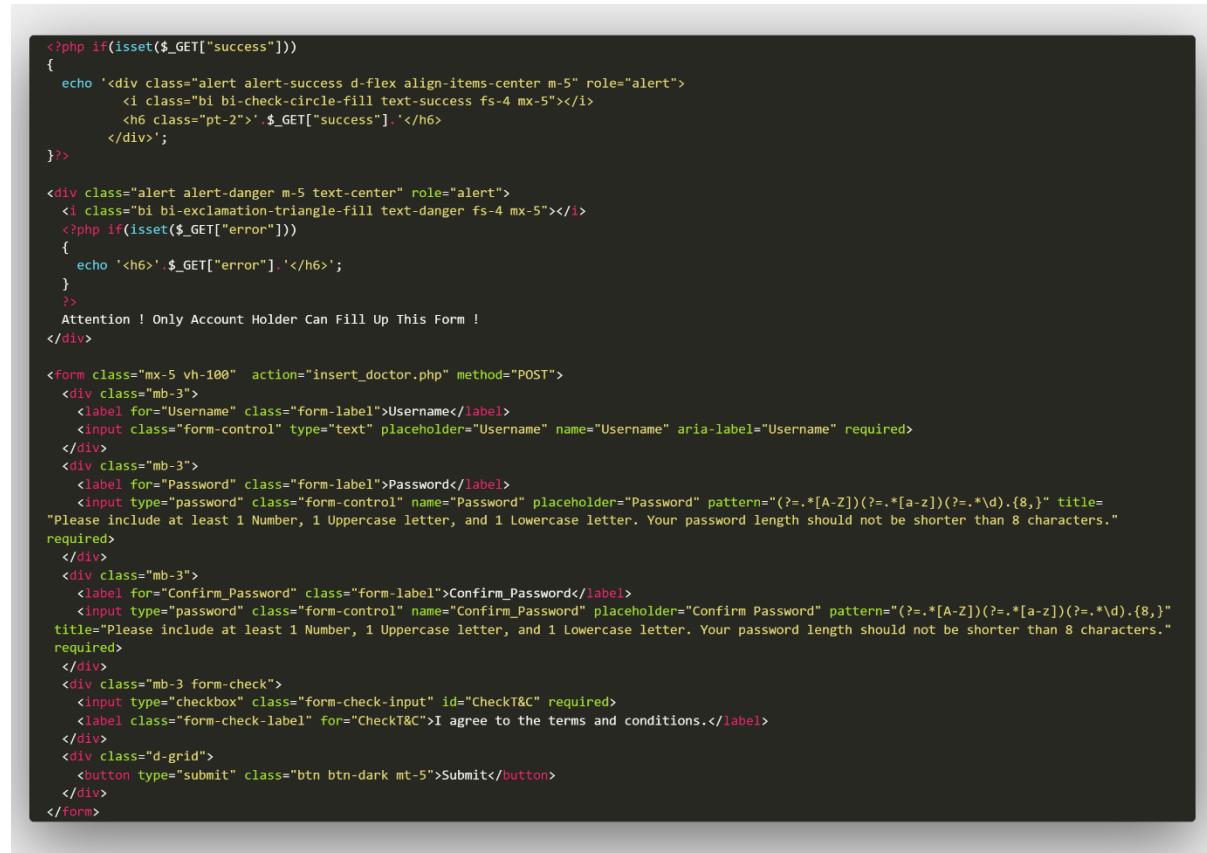


Figure 143: Create New Doctor Account Form and alert message

This is the HTML form for registering a new doctor to the doctor dashboard. This form's input fields are username, password, and confirm password. These input fields have certain limitations. Passwords must be at least eight characters long and include at least one number, one capital letter,

and one lowercase letter. Since this form utilizes the POST method, the code in insert doctor.php will be executed anytime the submit button is pressed.

```
$password = md5($_POST['Password']);
$sql="INSERT INTO `tbl_doctor` (Name, Contact_number, Specialist_category, Username, Password)
VALUES
('To Be Updated', 'To Be Updated', 'To Be Updated','$_POST[Username]','$password');

$check_existing = mysqli_query($conn,"SELECT * FROM `tbl_doctor` WHERE Username = '$_POST[Username]'");
$exist_rows = mysqli_num_rows($check_existing);

if (!$exist_rows == 0)
{
    header("Location: create_doctor.php?error= Username already taken! Please choose another username.");
}
else
if ($_POST['Password'] != $_POST['Confirm_Password'])
{
    header("Location: create_doctor.php?error= Please ensure your password and confirm password are matched.");
}
else
{
    mysqli_query($conn,$sql);
    header("Location: create_doctor.php?success= Doctor Account Successfully Created !");
}
```

Figure 144: insert\_admin.php

Administrators may use the create new doctor account tool to add new doctor accounts to the database. The create new account form will take the username, password, and confirm password via the POST method. After the form is submitted, the doctor's information will be obtained using the PHP POST method. Using the md5() function, the doctor's password will be hashed and encrypted. Additionally, if the password and confirm password do not match, the doctor will get an error message. The SQL INSERT INTO command is used to create column names and values that will be stored in the database to insert the doctor. The query will be executed using PHP's mysqli query() function. Finally, after the doctor account have been added to the database, a warning or success message will show above the form. Messages will be sent over the URL using the PHP header () function.

## 18.7.2 Read – HTML Form, PHP

### Search Treatment and Services

```
$query = "SELECT * FROM `tbl_medical_services` ";
$result = mysqli_query($conn, $query);

while($row = mysqli_fetch_array($result))
{
    >>
    <tr>
        <td class="text-center" >
            <a href='view_services.php?ServID=<?php echo $row['ServID'] ?>&' class="text-decoration-none text-dark" target="_blank">
                <?php echo $row['Title']?>
            </a>
        </td>
        <td class="text-center" >
            <a href='view_services.php?ServID=<?php echo $row['ServID'] ?>&' class="text-decoration-none text-dark" target="_blank">
                <?php echo $row['Specialist_category']?>
            </a>
        </td>
        <td class="text-center" >
            <a href='view_services.php?ServID=<?php echo $row['ServID'] ?>&' class="text-decoration-none text-dark" target="_blank">
                <?php echo $row['ConsultationFees']?>
            </a>
        </td>
    </tr>
```

Figure 145: Treatment and Services Table

The admin may use this search treatment and services option to look for and manage treatment and services in the online admin dashboard. DataTables JavaScript will sort the search result after admin entered search value. With a while loop wrapped around an HTML form, all the treatment and services results will be shown. If the administrator wishes to view, update, or delete treatment and services, admin can go to the respective row of treatment and services. To view treatment and services detail, the anchor tag will lead admin to the view treatment and services page.

## View Treatment and Services Details

```
<?php
$ServID = $_GET["ServID"];

$sql = "SELECT *
FROM `tbl_medical_services`
WHERE `ServID` = '$ServID'";
$result = mysqli_query($conn, $sql);
while($row = mysqli_fetch_array($result))
{
?>
<div class="row justify-content-between mb-5">
<div class="col-9">
<div class="mb-5">
<h3 class="card-title"><?php echo $row['Title'] ?></h3>
<p class="card-text"><?php echo $row['Description'] ?></p>
</div>
<div class="mb-5">
<h5 class="card-title">Consultation Fee Range</h5>
<p class="card-text"><?php echo $row['ConsultationFees'] ?></p>
</div>
<div class="mb-5">
<h5 class="card-title">Specialist Category</h5>
<p class="card-text"><?php echo $row['Specialist_category'] ?></p>
</div>
<div class="mb-5">

</div>
</div>
<div class="col-2 g-5 px-5 mx-0">
<div class="col m-5">
<a href="treatment_services.php">
<button type="button" class="btn btn-dark shadow"><i class="bi bi-search-heart-fill col-icon"></i>SEARCH</button>
</a>
</div>
<div class="col m-5">
<form action="update_services_form.php" method="post">
<input type="hidden" name="ServID" value="<?php echo $row['ServID']?>">
<button type="submit" class="btn btn-dark shadow" name="update"><i class="bi bi-pen-fill col-icon"></i>UPDATE</button>
</form>
</div>
<div class="col m-5">
<form action="delete_services.php" method="post">
<input type="hidden" name="ServID" value="<?php echo $row['ServID']?>">
<button type="submit" class="btn btn-dark shadow" name="delete"><i class="bi bi-trash-fill col-icon"></i>DELETE</button>
</form>
</div>
<?php
}
mysqli_close($conn);
?>
```

Figure 146: View Treatment and Services

This view treatment and services will get the Services ID with the GET method. Then the query will select the treatment and services with mysqli\_query() function. After that, mysqli\_fetch\_array() function is used to get the associative array of query result. The information is wrapped with the while loop which information is shown using the associative array.

## Total Users

```
<?php  
$sql_cus ="SELECT COUNT(*) AS num FROM `tbl_account`";  
$sql_admin ="SELECT COUNT(*) AS num FROM `tbl_admin_acc`";  
$sql_doc ="SELECT COUNT(*) AS num FROM `tbl_doctor`";  
$sql_staff ="SELECT COUNT(*) AS num FROM `tbl_staff_acc`";  
  
$result_cus = mysqli_query($conn, $sql_cus);  
$result_admin = mysqli_query($conn, $sql_admin);  
$result_doc = mysqli_query($conn, $sql_doc);  
$result_staff = mysqli_query($conn, $sql_staff);  
  
$cus = mysqli_fetch_array($result_cus);  
$admin = mysqli_fetch_array($result_admin);  
$doc = mysqli_fetch_array($result_doc);  
$staff = mysqli_fetch_array($result_staff);  
  
$total = $cus['num'] + $admin['num'] + $doc['num'] + $staff['num'];  
  
echo '<p card-text>' . $total . '</p>'  
;  
?>
```

Figure 147: admin\_dashboard.php

The PHP code above calculates the total users in the system. SELECT statement is used to select all number of rows from the account, admin, doctor, and staff tables. COUNT (\*) function counts number of rows in the table. The mysqli\_query() function executes a database query with connection and query parameter. The mysqli\_fetch\_array() function returns an associative array. Number of each type of user is gotten by using the associative arrays and are added together.

## Percentage of Each Type of User

```
<div class="col-6">
    <div class="card text-center bg-opacity-50 shadow-lg bg-info">
        <div class="card-body">
            <?php
                $c = $cus['num'] / $total * 100;
                $c = intval($c);
                echo '<h5 class="card-title">' . $c . '%</h5>';
            ?>
            <p class="card-text">Customer</p>
        </div>
    </div>
    <div class="col-6">
        <div class="card text-center shadow-lg bg-opacity-50 bg-warning">
            <div class="card-body">
                <?php
                    $a = $admin['num'] / $total * 100;
                    $a = intval($a);
                    echo '<h5 class="card-title">' . $a . '%</h5>';
                ?>
                <p class="card-text">Admin</p>
            </div>
        </div>
    </div>
    <div class="col-6">
        <div class="card text-center shadow-lg bg-opacity-50 bg-danger">
            <div class="card-body">
                <?php
                    $s = $staff['num'] / $total * 100;
                    $s = intval($s);
                    echo '<h5 class="card-title">' . $s . '%</h5>';
                ?>
                <p class="card-text">Staff</p>
            </div>
        </div>
    </div>
    <div class="col-6">
        <div class="card text-center shadow-lg bg-opacity-50 bg-primary">
            <div class="card-body">
                <?php
                    $d = $doc['num'] / $total * 100;
                    $d = intval($d);
                    echo '<h5 class="card-title">' . $d . '%</h5>';
                ?>
                <p class="card-text">Doctor</p>
            </div>
        </div>
    </div>

```

Figure 148: admin\_dashboard.php

The PHP code above shows how percentage of each type of user is calculated. The number of each type of user is retrieved using the associative array. This number is divided by the total number of user and multiply 100 to get the percentage. Intval() function is used to get the integer of the percentage.

## Total Sales Chart

```
<div class="row row-cols-2 g-4 mx-4">
  <div class="col-8">
    <div class="card shadow bg-white rounded-3">
      <div class="card-body m-4">
        <h5 class="card-title">Total Sales</h5>
        <canvas id="totalSalesChart"></canvas>
      </div>
    </div>
  </div>
```

Figure 149: admin\_dashboard.php

This is the white box for the bar chart of the total sales of 2021 and 2022. The canvas is included within the div so the chart will be placed within the white box.

```

// total sales

$sql_2021 = "SELECT MONTH(`Payment_date`) AS Month , SUM
(`Payment_total`) AS Sales
FROM `tbl_payment`
WHERE YEAR(`Payment_date`) = 2021
GROUP BY MONTH(`Payment_date`)
ORDER BY MONTH(`Payment_date`);"

$result_1 = mysqli_query($conn, $sql_2021);
$row_1 = mysqli_fetch_array($result_1);
$total_1 = array();

foreach ($result_1 as $row_1)
{
    array_push($total_1, $row_1['Sales']);
}
$Sales2021 = implode(',', $total_1);

$sql_2022 = "SELECT MONTH(`Payment_date`) AS Month , SUM
(`Payment_total`) AS Sales
FROM `tbl_payment`
WHERE YEAR(`Payment_date`) = 2022
GROUP BY MONTH(`Payment_date`)
ORDER BY MONTH(`Payment_date`);"

$result_2 = mysqli_query($conn, $sql_2022);
$row_2 = mysqli_fetch_array($result_2);
$total_2 = array();

foreach ($result_2 as $row_2)
{
    array_push($total_2, $row_2['Sales']);
}
$Sales2022 = implode(',', $total_2);

```

*Figure 150: admin\_dashbaord.php*

To get total sales of 2021 and 2022, SELECT keyword is used to select each month and total payment amount of each month. The month is retrieved with the MONTH () function getting the month part of the payment date. Total payment of the month can be gotten by using the SUM()

function returns total sum of the number of columns. To get total sales for every month in the year, the query is using GROUP BY statement to group rows with month. Moreover, the months is order by using the ORDER BY statement. The mysqli\_query() function executes a database query with connection and query parameter. The mysqli\_fetch\_array() function returns an associative array. The total\_1 and total\_2 arrays is created and used to store sales of every month with the array\_push() to insert sales of every month in the foreach loop. After getting all sales stored in the total array, the implode () function returns a string of the array elements with a comma as separator.

## Doctors' Gender Chart

```
<div class="col-4">
  <div class="card shadow bg-white rounded-3">
    <div class="card-body" style="height:59vh;">
      <h5 class="card-title p-3">Information About Clinic Doctors</h5>
      <div class="ms-3 mt-4" style="position: relative; width: 39vh;">
        <canvas id="genderChart"></canvas>
      </div>
    </div>
  </div>
</div>
```

Figure 151: admin\_dashboard.php

This is the white box for the doctor gender doughnut chart. The chart will be put inside the white box since the canvas is included within the div.

```
// gender chart

$sql = "SELECT `Gender`, COUNT(`Gender`) AS Amount
FROM `tbl_doctor`
GROUP BY `Gender`";

$result = mysqli_query($conn, $sql);
$row = mysqli_fetch_array($result);
$gender = array();

foreach ($result as $row)
{
    array_push($gender, $row['Amount']);
}

$male = $gender[0];
$female = $gender[1];
```

Figure 152: admin\_dashbaord.php

To get percentage of each gender, SELECT keyword is used to select each gender and amount of each gender. COUNT() function is used to count the number of each gender. The mysqli\_query() function executes a database query with connection and query parameter. The mysqli\_fetch\_array() function returns an associative array. The gender arrays are created and used to store amount of each gender with the array\_push() to insert gender and amount of that gender in the foreach loop.

## Appointment Status Chart

```
<div class="card-body m-4">
    <h5 class="card-title">Appointment Status</h5>
    <div style="position: relative; width: 40vh; margin-left: 26vh;">
        <canvas id="appChart"></canvas>
    </div>
</div>
```

Figure 153: report.php

The appointment status pie chart is shown in the white box. Since the canvas is located inside the div, the chart will be placed within the white box.

```
$sql_all = "SELECT COUNT(*) AS num
FROM `tbl_appointment`";
$result_all = mysqli_query($conn, $sql_all);
$row_all = mysqli_fetch_array($result_all);
$all = $row_all['num'];

$sql_sch = "SELECT COUNT(*) AS num
FROM `tbl_appointment`
WHERE `App_date` >= CURDATE()";
$result_sch = mysqli_query($conn, $sql_sch);
$row_sch = mysqli_fetch_array($result_sch);
$scheduled = $row_sch['num'];

$sql_comp = "SELECT COUNT(*) AS num
FROM `tbl_appointment`
WHERE `App_date` < CURDATE()";
$result_comp = mysqli_query($conn, $sql_comp);
$row_comp = mysqli_fetch_array($result_comp);
$completed = $row_comp['num'];
```

Figure 154: report.php

Select keyword is used to select all appointments from the appointment table. To count the number of each gender, use the COUNT () function. CURDATE() function is used to get current

date. With the current date, appointment status of scheduled and completed can be differentiate. The mysqli query() function runs a database query using the specified connection and query parameter. An associative array is returned by the mysqli fetch array() function. Value of scheduled and completed is gotten with the associative array.

## Daily Average Sale Chart

```
<div class="card-body m-4">
    <h5 class="card-title">Daily Average Sale</h5>
    <div style="position: relative; width: 72vh; margin-left: 8vh;">
        <canvas id="avgSaleChart"></canvas>
    </div>
</div>
```

Figure 155: report.php

In the white box, admin can see the daily average sale polar area chart. The chart will be put inside the white box since the canvas is within the div.

```
$sql_avg ="SELECT MONTH(`Payment_date`), SUM(`Payment_total`) / 30.42 AS avg_sale
FROM `tbl_payment`
GROUP by MONTH(`Payment_date`);

$result_avg = mysqli_query($conn, $sql_avg);
$row_avg = mysqli_fetch_array($result_avg);
$avg = array();

foreach($result_avg as $row_avg)
{
    array_push($avg, intval($row_avg['avg_sale']));
}
$avg = implode(',', $avg);
```

Figure 156: report.php

The Choose keyword is used to select each month and the total payment amount for each month to determine the average daily sale for 2021 and 2022. The MONTH () function is used to obtain the month part of the payment date. The SUM() function returns the total sum of the number of columns, which is then divided by 30.42, which is the average number of days in the month. The query uses the GROUP BY statement to group data by month to retrieve total sales for each month of the year. The mysqli query() function runs a database query using the given connection and

query parameter. An associative array is returned by the mysqli fetch array() method. The avg array is created and used to keep average daily sales for each month, with the array push() function included into the foreach loop to input average daily sales for each month. The implode () method produces a string of the array components with a comma as separator after collecting all average daily sales stored in the avg array.

## Top 10 Doctors

```
<?php
$sql_top = "SELECT `tbl_doctor`.* , SUM(`tbl_payment`.`Payment_total`) AS sales
FROM `tbl_payment`
INNER JOIN `tbl_appointment` ON `tbl_payment`.`AppID` = `tbl_appointment`.`AppID`
INNER JOIN `tbl_doctor` ON `tbl_appointment`.`DocID` = `tbl_doctor`.`DocID`
GROUP BY `tbl_doctor`.`DocID`
ORDER BY SUM(`tbl_payment`.`Payment_total`) DESC
LIMIT 10";

$result_top = mysqli_query($conn, $sql_top);

$num = 1;
while($row_serv = mysqli_fetch_array($result_top))
{
?>
<div class="col-4">
<div class="mt-4 mb-2">
<div class="row">
    <h3 class="card-title text-info"><?php echo $num ?></h3>
</div>
</div>
</div>

<div class="col-6">
<div class="mt-4 mb-2">
<div class="row">
    <h6 class="card-title"><?php echo $row_serv['Name'] ?></h6>
    <p card-text>RM <?php echo $row_serv['sales'] ?></p>
</div>
</div>
</div>

<?php
    $num++;
}
?>
```

Figure 157: Top 10 Doctors

In the SQL query above, all columns of doctor table are selected and the sales in payment table. INNER JOIN statement is used connect the primary key of a few tables to get the doctor table, appointment table and payment table connected. The results are group with the doctor by using GROUP BY statement. ORDER BY statement is used to organize the sales from more to less. LIMIT statement will get the first 10 row of result. Using mysqli\_query() function, the SQL query is executed and mysqli\_fetch\_array() will get the result in associative array. The while loop will wrap the column to display the name and sales of each doctor. A num variable is used to increment the rank of doctor.

### 18.7.3 Update – HTML Form, PHP

#### Update Treatment and Services

```
<form action="update_services_form.php" method="post">
  <input type="hidden" name="ServID" value="php echo $row['ServID'];?">
  <td class="text-center" ><button type="submit" class="btn btn-outline-dark" name="update">Update</button></td>
</form>
```

Figure 158: Update Treatment and Services button

This is the update button in the row of Treatment and services while admin search for treatment and services.

```

<?php
if(isset($_POST['update'])){
    $servID = $_POST["ServID"];

    $sql = "SELECT *
    FROM `tbl_medical_services`
    WHERE `ServID` = '$servID'";
    $result = mysqli_query($conn, $sql);
    while($row = mysqli_fetch_array($result))
    {

    >>>
        <div class="col-12">
            <div class="row justify-content-center text-center align-items-center">
                <div class="col-5">
                    
            </div>
        </div>

        <form class="mx-5 vh-100" action="update_services.php" method="post">
            <input type="Hidden" name="ServID" value="<?php echo $row['ServID'] ??">

            <div class="mb-3">
                <label for="Title" class="form-label">Title</label>
                <input class="form-control" type="text" placeholder="Title" name="Title"
aria-label="Title" value="<?php echo $row['Title'] ??" required>
            </div>
            <div class="mb-3">
                <label for="Description" class="form-label">Description</label>
                <textarea class="form-control" placeholder="Description" name="Description"
aria-label="Description" style="height: 100px" required><?php echo $row[
'Description'] ?</textarea>
            </div>
            <div class="mb-3">
                <label for="Specialist_category" class="form-label">Specialist Category</label>
                <input class="form-control" type="text" placeholder="Specialist Category" name=
"Specialist_category" aria-label="Specialist Category" value="<?php echo $row[
'Specialist_category'] ??" required>
            </div>
            <div class="mb-3">
                <label for="Consultation_Fees" class="form-label">Consultation Fee Range</label>
                <select class="form-select" aria-label="Consultation Fee" name=
"Consultation_Fees" required>
                    <option value="RM30 - RM50">
<?php if($row['ConsultationFees'] == "RM30 - RM50"){ ?>
selected="selected"
<?php } ?>
                    >RM30 - RM50</option>

                    <option value="RM51 - RM89">
<?php if($row['ConsultationFees'] == "RM51 - RM89"){ ?>
selected="selected"
<?php } ?>
                    >RM51 - RM89</option>

                    <option value="RM91 - RM125">
<?php if($row['ConsultationFees'] == "RM91 - RM125"){ ?>
selected="selected"
<?php } ?>
                    >RM91 - RM125</option>

                    <option value="RM126 - RM235">
<?php if($row['ConsultationFees'] == "RM126 - RM235"){ ?>
selected="selected"
<?php } ?>
                    >RM126 - RM235</option>
                </select>
            </div>
            <div class="d-grid">
                <button type="submit" class="btn btn-dark mt-5" name="update">Submit</button>
            </div>
        </form>
    </?php
    }
}

mysqli_close($conn);
?>

```

Figure 159: Update Treatment and Services Form

The PHP code above the HTML form. If the update button is submitted, the service ID of the treatment and services to identify which services. Select keyword select all details from the treatment and services. Where statement allows the query to select one of the rows. Mysqli\_query() function will execute the SQL query with the connection given. Mysqli\_fetch\_array() will get the associative array of the result. The update treatment and services form are wrapped with the while loop. Associative array is used to get each value from the phpMyAdmin. Once the admin clicks the submit button, the information will be passed to the update\_services.php with the POST method.

```

function validate($data)
{
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

$title = validate($_POST['Title']);
$description = validate($_POST['Description']);
$specialist_category = validate($_POST['Specialist_category']);
$consultation_fees = validate($_POST['Consultation_Fees']);

$check_existing = mysqli_query($conn, "SELECT * FROM `tbl_medical_services` WHERE ServID = '".$_POST[ServID]'");
$exist_rows = mysqli_num_rows($check_existing);

if (!$exist_rows == 1)
{
    header("Location: update_services_form.php?error= Please Try Again !");
}
else
{
    $sql = "UPDATE `tbl_medical_services` SET
    `Title` = '$title',
    `Description` = '$description',
    `Specialist_category` = '$specialist_category',
    `ConsultationFees` = '$consultation_fees'
    WHERE `ServID` = '".$_POST[ServID]"";

    if(mysqli_query($conn,$sql))
    {
        header("Location: update_services_form.php?success= Treatment and Services Successfully Updated !");
    }
    else
    {
        header("Location:update_services_form.php?error= Please Try Again !");
    }
}

```

*Figure 160: update\_services.php*

The validate function will validate the data by trimming the data, strip slashes from the data and transform some predefined HTML characters into HTML entities. All value gotten from the update treatment and services form will be validated. There will be a mysqli\_query() function and mysqli\_num\_rows() function to check existing treatment and services in the database. If the row of treatment and services of the Services ID given is not only one, but there will also be a warning message to notify the admin, else the treatment services will be updated, and a success message will be shown.

#### 18.7.4 Delete – HTML Form, PHP

##### Delete Treatment and Services

```
<form action="delete_services.php" method="post">
    <input type="hidden" name="ServID" value="<?php echo $row['ServID']?>">
    <td class="text-center" ><button type="submit" class="btn btn-outline-dark" name="delete">Remove</button></td>
</form>
```

Figure 161: Delete Treatment and Services

This is the delete button shown up in the treatment and services table when admin search for treatment and services. When admin click on this button, the treatment and services will be deleted.

```
if(isset($_POST['delete']))
{
    $check_existing = mysqli_query($conn, "SELECT * FROM `tbl_medical_services` WHERE ServID = '$_POST[ServID]'");
    $exist_rows = mysqli_num_rows($check_existing);

    if (!$exist_rows == 1)
    {
        header("Location: treatment_services.php?error= Please Try Again !");
    }
    else
    {
        $sql = "DELETE FROM `tbl_medical_services` WHERE `ServID` = '$_POST[ServID]'";
        mysqli_query($conn,$sql);
        header("Location: treatment_services.php?success= Treatment and Services Successfully Deleted !");
    }
}
```

Figure 162: Delete Treatment and Services

This is the delete\_services.php which the PHP code to delete the treatment and services is placed. If the delete button is clicked, the PHP code will be executed. First, the code will check if the Services ID exist in the system, and is the Services ID is unique. If Services ID is unique, the Treatment and Services will be deleted using the DELETE FROM statement, and WHERE statement to specify which Services to delete. Mysqli\_query() helps to execute the SQL query with a database connection given. There will be a message to notice the admin if the treatment and services is deleted successfully or not.

### 18.7.5 Login – HTML Form, PHP

```
<?php if (isset($_GET['error']))  
{  
    echo  
'<div class="alert alert-danger m-5 text-center" role="alert">' .  
        $_GET['error'] .  
    '</div>';  
} ?>  
  
<form class="mx-5" action="admin_logincheck.php" method="post">  
    <div class="mb-3">  
        <label for="Username" class="form-label">Username</label>  
        <input class="form-control" type="text" placeholder=  
    "Username" name="Username" aria-label="Username">  
    </div>  
    <div class="mb-3">  
        <label for="Password" class="form-label">Password</label>  
        <input type="password" class="form-control" placeholder=  
    "Password" name="Password">  
    </div>  
    <div class="d-grid">  
        <button type="submit" class="btn btn-dark mt-5 mb-5 rounded">  
    Submit</button>  
    </div>  
</form>
```

Figure 163: admin\_login.php

This is the login form of admin dashboard. There is PHP if statement above the form. The PHP code will display the error message once there is error received using the GET method.

```
function validate($data)
{
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

$username = validate($_POST['Username']);
$password = validate($_POST['Password']);
```

Figure 164: admin\_logincheck.php

After admin submit the login form, PHP code in the admin\_logincheck.php will be executed to check the login credentials. This validate function contains PHP functions of trim(), stripslashes() and htmlspecialchars(). Trim() function eliminate whitespaces and other preset characters off both sides of a string. stripslashes() remove backslashes. htmlspecialchars() function transform some preset characters into HTML entities. This validate function will return clean username and password.

```

if (empty($username) && empty($password))
{
    header("Location: admin_login.php?error=Fields cannot be empty!");
    exit();
}
else if (empty($username))
{
    header("Location: admin_login.php?error=Username is required!");
    exit();
}
else if(empty($password))
{
    header("Location: admin_login.php?error=Password is required!");
    exit();
}
else
{
    $password = md5($password);
    $sql = "SELECT * FROM `tbl_admin_acc` WHERE Username = '$username' AND
Password = '$password';

$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) == 1)
{
    $row = mysqli_fetch_assoc($result);

    if ($row['Username'] === $username && $row['Password'] ===
$password)
    {
        $_SESSION['Admin_Username'] = $row['Username'];

        header("Location: admin_dashboard.php");
    }
    else
    {
        header(
"Location: admin_login.php?error= Incorrect Username or Password!");
        exit();
    }
}
else
{
    header(
"Location: admin_login.php?error= Incorrect Username or Password!");
    exit();
}
}

```

Figure 165: admin\_logincheck.php

This PHP code validates the login credentials of admin. When there are empty field in username or password, an error message will be shown on the screen by using the header function. The error message was sent by using header () function to send a HTTP header. In the login page, PHP code will be displayed when there is error send through the HTTP header. Furthermore, password will be hashed by using the md5() function. This will help to encrypt the password. An SQL statement is written to match login credentials with the admin accounts in database. All admin information of the matched username and password are extracted from the tbl\_admin\_acc table. The mysqli\_query() function executes a database query with connection and query parameter. The mysqli\_num\_rows() return the number of rows in the result from mysqli\_query() function. In this case, if there is only one admin matches the username and password, the mysqli\_fetch\_assoc() function returns an associative array. If username and password entered by admin matches the username and password in database, admin will be redirected to the overview of admin dashboard. If the username or password were incorrect, there will be an error message to notice the admin.

## 18.7.6 Signup – HTML Form, PHP

### Create New Admin Account

```
<?php if(isset($_GET["success"])) {
    echo '<div class="alert alert-success d-flex align-items-center m-5" role="alert">
        <i class="bi bi-check-circle-fill text-success fs-4 mx-5"></i>
        <h6 class="pt-2">' . $_GET["success"] . '</h6>
    </div>';
} ?>

<div class="alert alert-danger m-5 text-center" role="alert">
    <i class="bi bi-exclamation-triangle-fill text-danger fs-4 mx-5"></i>
    <?php if(isset($_GET["error"])) {
        echo '<h6>' . $_GET["error"] . '</h6>';
    }
    ?>
    Attention ! Only Account Holder Can Fill Up This Form !
</div>

<form class="mx-5 vh-100" action="insert_admin.php" method="POST">
    <div class="mb-3">
        <label for="Username" class="form-label">Username</label>
        <input class="form-control" type="text" placeholder="Username" name="Username" aria-label="Username" required>
    </div>
    <div class="mb-3">
        <label for="Password" class="form-label">Password</label>
        <input type="password" class="form-control" name="Password" placeholder="Password" pattern="(?=.*[A-Z])(?=.*[a-z])(?=.*\d).{8,}" title="Please include at least 1 Number, 1 Uppercase letter, and 1 Lowercase letter. Your password length should not be shorter than 8 character s." required>
    </div>
    <div class="mb-3">
        <label for="Confirm_Password" class="form-label">Confirm Password</label>
        <input type="password" class="form-control" name="Confirm_Password" placeholder="Confirm Password" pattern="(?=.*[A-Z])(?=.*[a-z])(?=.*\d).{8,}" title="Please include at least 1 Number, 1 Uppercase letter, and 1 Lowercase letter. Your password length should not be shorter than 8 character s." required>
    </div>
    <div class="mb-3 form-check">
        <input type="checkbox" class="form-check-input" id="CheckT&C" required>
        <label class="form-check-label" for="CheckT&C">I agree to the terms and conditions.</label>
    </div>
    <div class="d-grid">
        <button type="submit" class="btn btn-dark mt-5">Submit</button>
    </div>
</form>
```

Figure 166: Create New Admin Account Form and alert message

This is the HTML form for creating new admin account to the admin dashboard. The username, password and confirm password are the input field of this form. There are restriction in these input fields. Password are limited to at least 8 characters and contains at least 1 number, 1 uppercase letter and 1 lowercase letter. The code in insert\_admin.php will be run whenever the submit button is hit since this form uses the POST method.

```

$password = md5($_POST['Password']);
$sql="INSERT INTO `tbl_admin_acc` (Username, Password)
VALUES
('$_POST[Username]','$password');

$check_existing = mysqli_query($conn,"SELECT * FROM `tbl_admin_acc` WHERE Username = '$_POST[Username]'");
$exist_rows = mysqli_num_rows($check_existing);

if (!$exist_rows == 0)
{
    header("Location: create_admin.php?error= Username already taken! Please choose another username.");
} else
if ($_POST['Password'] != $_POST['Confirm_Password'])
{
    header("Location: create_admin.php?error= Please ensure your password and confirm password are matched.");
}
else
{
    mysqli_query($conn,$sql);
    header("Location: create_admin.php?success= Admin Account Successfully Created !");
}

```

*Figure 167: insert\_admin.php*

This create new admin account function allows administrators to add new admin account to the database. The username, password and confirm password will be accepted by the create new account form using the POST technique. The admin information will be received using the PHP POST method after the form is submitted. Password of the admin will be hashed and encrypted using the md5() function. Furthermore, if the password and confirm password does not match, there will also be an error message to the admin. To insert the admin, the SQL INSERT INTO statement is used to establish column names and values that will be placed into the database. The mysqli\_query() function in PHP will run the query. Finally, the warning or success message will appear above the form when admin account have been added to the database. Messages will be sent by using PHP header() function which send through URL.

## 18.7.8 Self-created CSS and JavaScript

### Search and Sort Treatment and Services

```
<script>
$(document).ready(function() {
    $('#search-treatment').DataTable( {
        dom: 'lBfrtip',
        buttons: [ 'copy', 'csv', 'excel', 'pdf', 'print'
    ],
    pagingType: 'full',
    initComplete: function () {
        var btns = $('.dt-button');
        btns.addClass('btn btn-dark mx-3');
        btns.removeClass('dt-button');
    }
} );
} );
</script>
```

Figure 168: DataTable Plugin for Search Treatment Table

Code inside \$(document). ready () will only be called once the page's Document Object Model (DOM) is ready to execute JavaScript code. DOM is set to lBfrtrip which allow admin to determine how many rows will be showed on the screen with the drop-down menu. Export buttons like Copy, CSV, Excel, PDF, and Print is provided. pagingType is set to full where the table will auto fit the page size. initComplete () function means the code within will only be executed after the table's data successfully loaded. Variable btns is used to select the class name. By using the variable, the class name of Bootstrap CSS is added into the button to style the button. After that, the class name “dt-button” will be removed.

## Total Sales Chart

```
const Sales2021 = [<?php echo $Sales2021 ?>];
const Sales2022 = [<?php echo $Sales2022 ?>];

const labels = [
    'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct',
    'Nov', 'Dec'
];
const data = {
    labels: labels,
    datasets: [
        {
            label: '2021',
            borderColor: '#CFBAF0',
            backgroundColor: ['#FBF8CC', '#FDE4CF', '#FFCFD2', '#F1C0E8',
                '#CFBAF0', '#A3C4F3', '#90DBF4', '#8EECF5', '#98F5E1', '#B9FBC0',
                '#9BF6FF', '#BEE1E6'],
            data: Sales2021
        },
        {
            label: '2022',
            borderColor: '#A3C4F3',
            backgroundColor: ['#FBF8CC', '#FDE4CF', '#FFCFD2', '#F1C0E8',
                '#CFBAF0', '#A3C4F3', '#90DBF4', '#8EECF5', '#98F5E1', '#B9FBC0',
                '#9BF6FF', '#BEE1E6'],
            data: Sales2022
        }
    ]
};
const config = {
    type: 'bar',
    data: data,
    options: {}
};
const totalSalesChart = new Chart(
document.getElementById('totalSalesChart'),
config
);
```

Figure 169: admin\_dashbaord.php

Figure above shows the JavaScript of chart's data. Sales of 2021 and 2022 is retrieved from the PHP string. In the code above, labels of each month are stored in a constant array. Label, border color, background color and data are stored in a constant object. In config constant object, chart type is set as bar chart and the data is assign with the data constant object. totalSalesChart constant object has a new keyword to create Chart object. In the Chart object, getElementById() is used to get element with the totalSalesChart id, config constant object is also included.

## Doctors' Gender Chart

```
const female = [<?php echo $female ?>];
const male = [<?php echo $male ?>];

const stats_labels = ['Male', 'Female'];
const stats_data = {
    labels: stats_labels,
    datasets: [
        {
            label: 'Doctors',
            backgroundColor: ['#abc4ff', '#fbb1bd'],
            data: [male, female]
        }]
};
const stats_config = {
    type: 'doughnut',
    data: stats_data,
    options: {}
};
const genderChart = new Chart(
document.getElementById('genderChart'),
stats_config
);
```

Figure 170: admin\_dashbaord.php

The JavaScript for the data in the chart is shown in the image above. The PHP array is used to obtain the number of females and males. Female and male are saved in a stat labels constant array. A constant object stores the label, background color, and data. The chart type is set to doughnut chart in the config constant object, and the data is assigned to the stats data constant object. A new keyword for creating Chart objects has been added to the genderChart constant object. getElementById() is used in the Chart object to get the element with the genderChart id, and a stats config constant object is also provided.

## Appointment Status Chart

```
const scheduled = [<?php echo $scheduled ?>];
const completed = [<?php echo $completed ?>];

const app_labels = [
    'Scheduled', 'Completed'
];

const app_data = {
    labels: app_labels,
    datasets: [
        {
            label: 'Status',
            backgroundColor: ['#84ffc9', '#aab2ff'],
            data: [scheduled, completed]
        }]
};
const app_config = {
    type: 'pie',
    data: app_data,
    options: {}
};
const appChart = new Chart(
    document.getElementById('appChart'),
    app_config
);
```

Figure 171: Report.php

In the figure above, the JavaScript for the chart's data can be seen. The number of booked and completed appointments is obtained using the PHP variable. A constant array of app labels stores “Scheduled” and “Completed”. Appointment status the label, background color and data are stored in a constant object. The data is allocated to the app data constant object, and the chart type is pie chart. In the appChart constant object, a new keyword for constructing Chart objects has been introduced. The Chart object uses getElementById() to get the element with the appChart id, and there is also an app config constant object.

## Daily Average Sale Chart

```
const avg = [<?php echo $avg ?>];

const avg_labels = [
    'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'
];

const avg_data = {
    labels: avg_labels,
    datasets: [
        {
            label: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'],
            backgroundColor: ['#FBF8CC', '#FDE4CF', '#FFCFD2', '#F1C0E8', '#CFBAF0', '#A3C4F3',
                '#90DBF4', '#8EECF5', '#98F5E1', '#B9FBC0', '#9BF6FF', '#BEE1E6'],
            data: avg
        }
    ]
};

const config_avg = {
    type: 'polarArea',
    data: avg_data,
    options: {}
};

const avgSaleChart = new Chart(
document.getElementById('avgSaleChart'),
config_avg
);
```

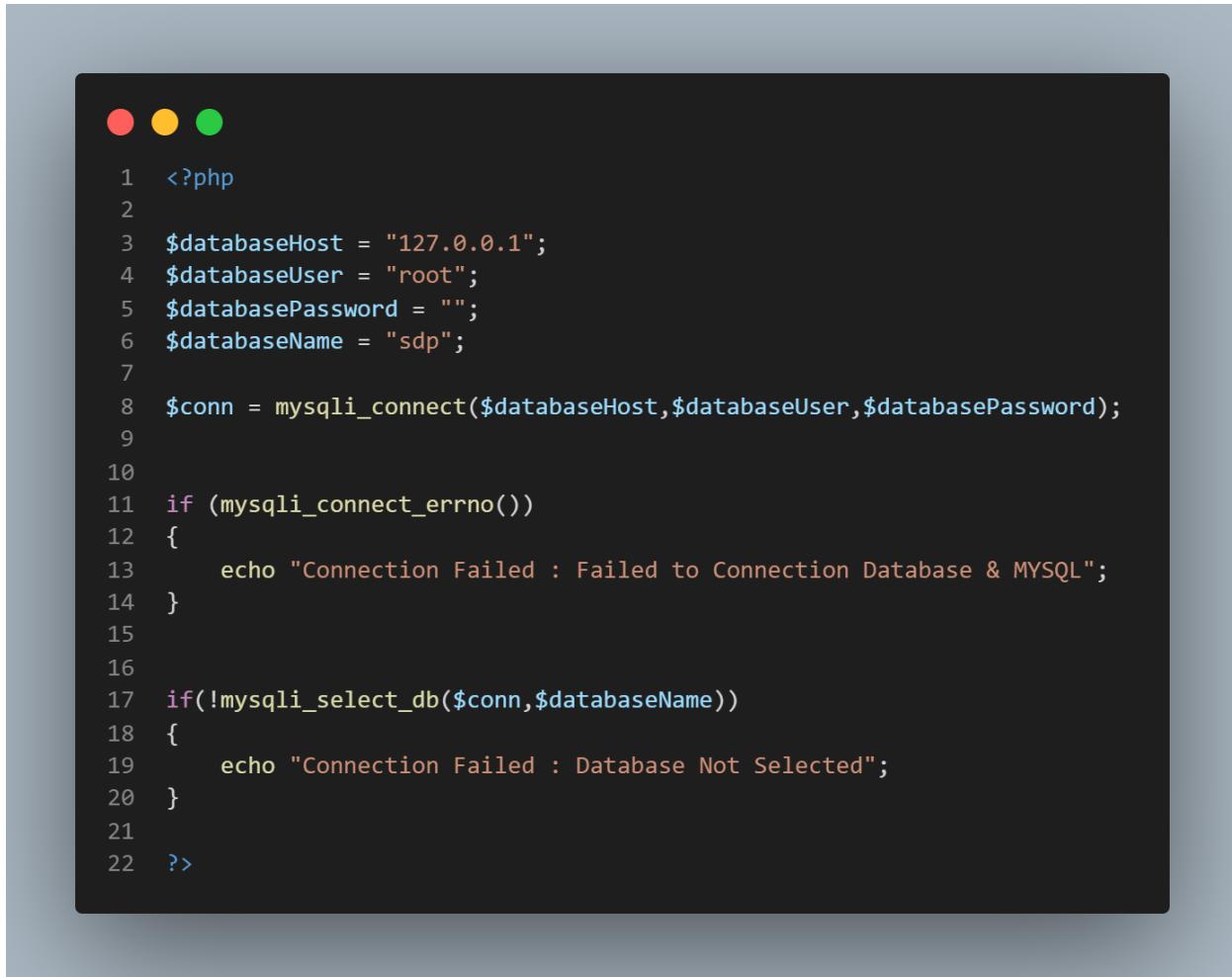
Figure 172: report.php

The JavaScript for the data in the chart is shown in the image above. The PHP string is used to get the average daily sales for 2021 and 2022. Each month's labels are saved in a constant array in the code above. A constant object stores the label, border color, background color, and data. The chart type is set to polar area chart in the config constant object, and the data is assigned to the data constant object. A new keyword for creating Chart objects has been added to the avgSaleChart constant object. The getElementById() method in the Chart object is used to get the element with the avgSaleChart id, and a config constant object is also added.

## 18.8 Doctor Module

### 18.8.1 Create – HTML Form, PHP

#### Create connection to the database



The screenshot shows a terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top. The terminal displays the following PHP code:

```
1 <?php
2
3 $databaseHost = "127.0.0.1";
4 $databaseUser = "root";
5 $databasePassword = "";
6 $databaseName = "sdp";
7
8 $conn = mysqli_connect($databaseHost,$databaseUser,$databasePassword);
9
10
11 if (mysqli_connect_errno())
12 {
13     echo "Connection Failed : Failed to Connection Database & MYSQL";
14 }
15
16
17 if(!mysqli_select_db($conn,$databaseName))
18 {
19     echo "Connection Failed : Database Not Selected";
20 }
21
22 ?>
```

Figure 173 Dr\_connection.php

Connect to the database server and notice user “Connection Failed” with reasons when the program failed to connect to the database server

## 18.8.2 Read – HTML Form, PHP

### Read Schedule



```
<?php
    require "connection/connection.php";
    $doctorid=$_SESSION['DocID'];
    $sql="SELECT appointment.AppointmentID,appointment.Acc_ID,tbl_account.Name,appointment.
    DoctorID,appointment.Date,appointment.Time
    FROM `appointment`INNER JOIN tbl_account ON appointment.Acc_ID=tbl_account.Acc_ID
    WHERE DoctorID=$doctorid and Date=CURRENT_DATE ";
    $records =mysqli_query($conn,$sql);

?
<?php
    if ($records->num_rows>0){
        while($row =mysqli_fetch_array($records))
    {
        echo'<div class="row col-9">
        <div class="card shadow bg-info rounded-3">
        <br>
        <h4 class="card_title">AppointmentID: '.$row['AppointmentID'].'</h5>
        <h5 class="card_title">PatientID : '.$row['Acc_ID'].'</h4>
        <h5 class="card_title">Patient Name : '.$row['Name'].'</h4>
        <h5 class="card_title">Time : '.$row['Time'].'</h4>
        </div>
        </div>
        <br>';
    }
}else{
    echo'<div class="row col-9">
    <div class="card shadow bg-success rounded-3" >
    <br><br>
    <h4 class="card_title">No Appointment For Today<br></h4>
    <br><br>
    </div>
    </div>
    ';
}
?>
```

Figure 174 Read Schedule Part on Dr\_Homepage.php

Connect to the database server at first and then acquire current doctor ID from session. Next select SQL statement to read data from the database. The SQL is about filtering all appointment based on Doctor ID on the current time date. Lastly, using a while loop to create card and print all the data on it, else if there are no data found the system will print “No Appointment for Today” instead.

## Read Profile Picture and Username



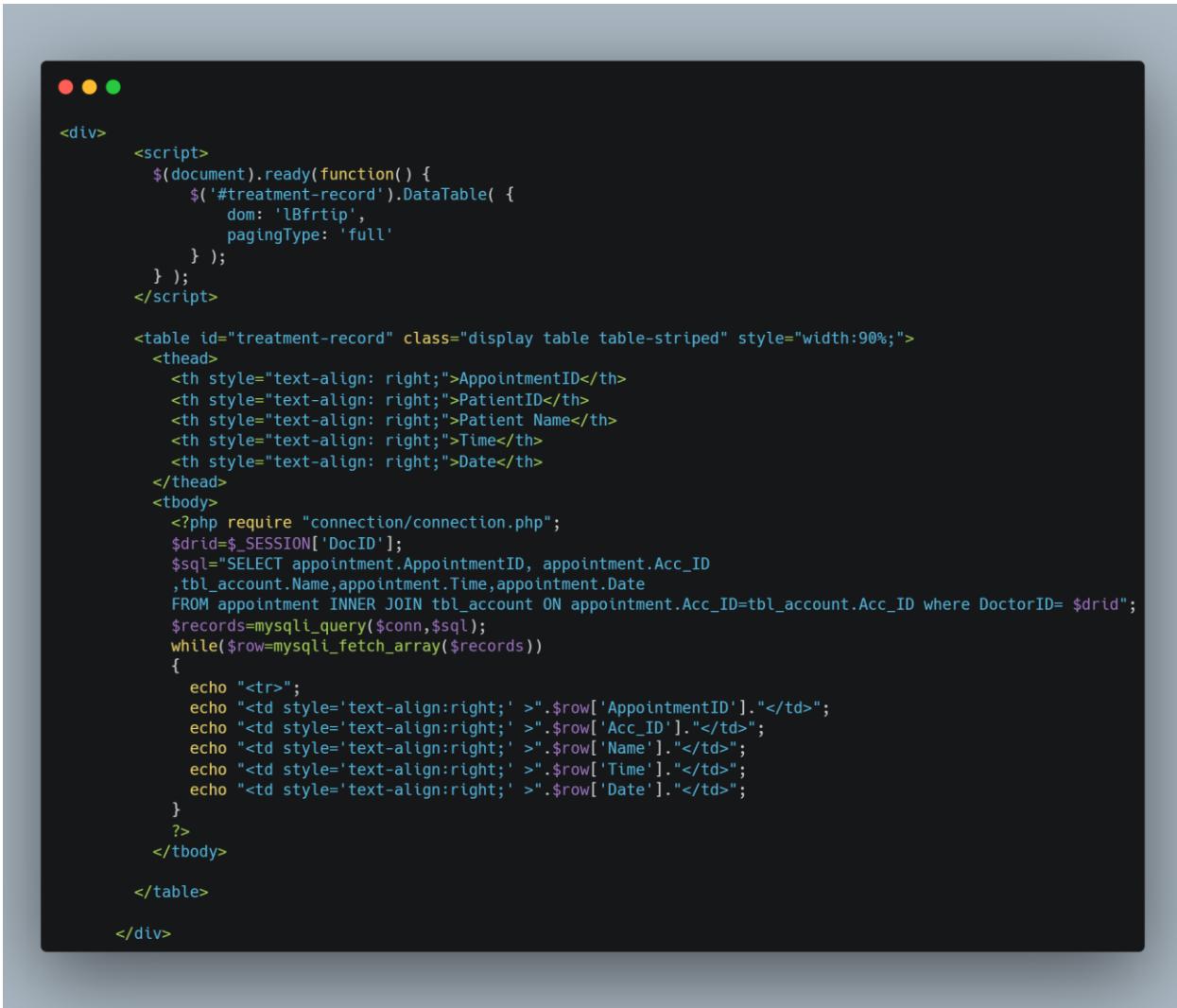
```
<?php
require "connection/connection.php";
$doctorid=$_SESSION['DocID'];
$sql02="SELECT * FROM `tbl_doctor` WHERE DocID=$doctorid ";
$records02 =mysqli_query($conn,$sql02);
$data=mysqli_fetch_array($records02);
?>

<div class="card shadow bg-white rounded-3">
<div class="card-body m-4">
<div class="text-center">
<?php
echo '';
echo '<br>';
?>
<h2>@<?=$_SESSION['username']?></h2>
</div>
</div>
</div>
```

Figure 175 Read Profile Picture and Username Part in Dr\_homepage.php

Connect to the database server at first and then acquire current doctor ID from session. Next select SQL statement to read data from the database. The SQL is about reading all data from the doctor table where the Doctor ID is equal to the session ID. Next will be creating a card and then echo the profile picture and username inside the card column.

## Read Appointment



The screenshot shows a code editor window with a dark theme. The code is written in PHP and displays a script block containing a jQuery DataTable initialization and a table element. The table has a header row with five columns: AppointmentID, PatientID, Patient Name, Time, and Date. The body of the table contains rows of data fetched from a database query. The code uses echo statements to print the table structure and data.

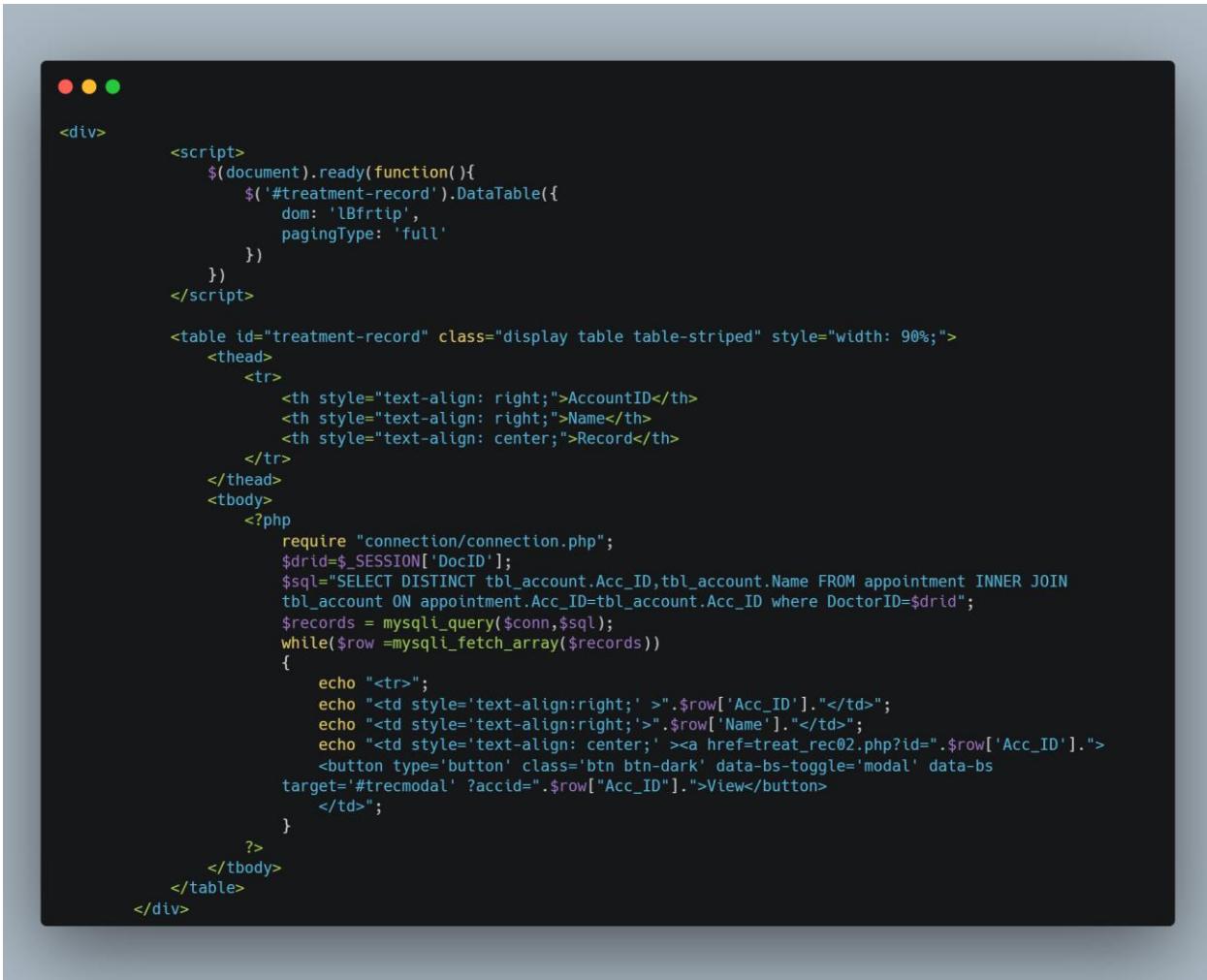
```
<div>
<script>
$(document).ready(function() {
    $('#treatment-record').DataTable( {
        dom: 'Lfrtip',
        pagingType: 'full'
    } );
} );
</script>

<table id="treatment-record" class="display table table-striped" style="width:90%; ">
<thead>
    <th style="text-align: right;">AppointmentID</th>
    <th style="text-align: right;">PatientID</th>
    <th style="text-align: right;">Patient Name</th>
    <th style="text-align: right;">Time</th>
    <th style="text-align: right;">Date</th>
</thead>
<tbody>
<?php require "connection/connection.php";
$drid=$_SESSION['DocID'];
$sql="SELECT appointment.AppointmentID, appointment.Acc_ID
,tbl_account.Name,appointment.Time,appointment.Date
FROM appointment INNER JOIN tbl_account ON appointment.Acc_ID=tbl_account.Acc_ID where DoctorID= $drid";
$records=mysqli_query($conn,$sql);
while($row=mysqli_fetch_array($records))
{
    echo "<tr>";
    echo "<td style='text-align:right;' >".$row['AppointmentID']. "</td>";
    echo "<td style='text-align:right;' >".$row['Acc_ID']. "</td>";
    echo "<td style='text-align:right;' >".$row['Name']. "</td>";
    echo "<td style='text-align:right;' >".$row['Time']. "</td>";
    echo "<td style='text-align:right;' >".$row['Date']. "</td>";
}
?>
</tbody>
</table>
</div>
```

Figure 176 Read Appointment Part at Dr\_appointment.php

At first, use the script tag to call the table function. After that, start the table element and fill in the content of the table. For the table header, all the text will be aligned on the right side. For the table body, the system will first read data from the database and then only echo it out on the table body. The data read in this table is all data about the appointment.

## Read Account



```
<div>
    <script>
        $(document).ready(function(){
            $('#treatment-record').DataTable({
                dom: 'lBfrtp',
                pagingType: 'full'
            })
        })
    </script>

    <table id="treatment-record" class="display table table-striped" style="width: 90%; ">
        <thead>
            <tr>
                <th style="text-align: right;">AccountID</th>
                <th style="text-align: right;">Name</th>
                <th style="text-align: center;">Record</th>
            </tr>
        </thead>
        <tbody>
            <?php
                require "connection/connection.php";
                $drid=$_SESSION['DocID'];
                $sql="SELECT DISTINCT tbl_account.Acc_ID,tbl_account.Name FROM appointment INNER JOIN
                tbl_account ON appointment.Acc_ID=tbl_account.Acc_ID where DoctorID=$drid";
                $records = mysqli_query($conn,$sql);
                while($row =mysqli_fetch_array($records))
                {
                    echo "<tr>";
                    echo "<td style='text-align:right;' >".$row['Acc_ID']."</td>";
                    echo "<td style='text-align:right;' >".$row['Name']. "</td>";
                    echo "<td style='text-align: center;' ><a href=treat_rec02.php?id=".$row['Acc_ID']. ">
                    <button type='button' class='btn btn-dark' data-bs-toggle='modal' data-bs-
                    target='#trecmodal' ?accid='".$row["Acc_ID"]. ">View</button>
                    </td>";
                }
            ?>
        </tbody>
    </table>
</div>
```

Figure 177 Read Account Part at Dr\_acc.php

Call the table function within the script tag. Create the table and set the table header alignment to right. Read data from the database acquiring Account ID, and Name. Echo the data in the table data tag with right alignment. For each row of data create a view button beside to direct the user to a more detail page.

## Read Account Appointment



```
<div>
    <script>
        $(document).ready(function(){
            $('#patient-record').DataTable({
                dom: 'LBfrtip',
                pagingType: 'full'
            })
        })
    </script>

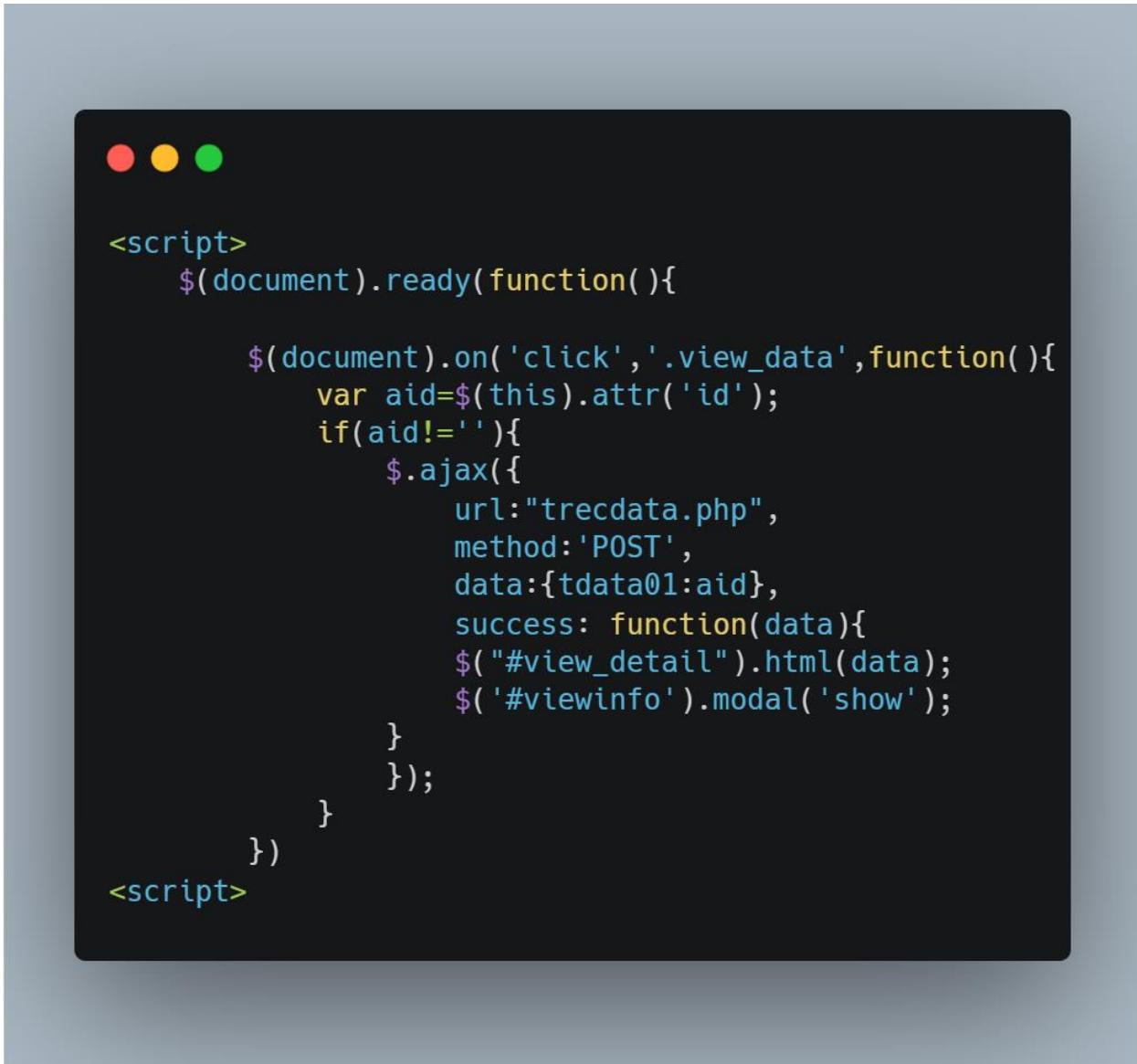
    <table id="patient-record" class="display table table-striped" style="width: 90%;">
        <thead>

            <tr>
                <th style="text-align: right;">AppointmentID</th>
                <th style="text-align: right;">Time</th>
                <th style="text-align: right;">Date</th>
                <th style="text-align: right;">View</th>
                <th style="text-align: right;">Edit</th>
            </tr>
        </thead>
        <tbody>
            <?php
                $sql01="SELECT * FROM `appointment` INNER JOIN tbl_account ON appointment.Acc_ID=tbl_account.Acc_ID WHERE
                    DoctorID=$doctorid and appointment.Acc_ID=$id";
                $records02=mysqli_query($conn,$sql01);
                $num=0;
                while($row =mysqli_fetch_array($records02))
                {
                    echo "<tr>";
                    echo "<td style='text-align:right;' >".$row['AppointmentID']."</td>";
                    echo "<td style='text-align:right;'>".$row['Time']."'</td>";
                    echo "<td style='text-align:right;'>".$row['Date']."'</td>";
                    echo "<td style='text-align: right;' >
                        <button id='".$row['AppointmentID']."' class='btn btn-dark view_data'>View</button>
                    </td>";
                    echo "<td style='text-align: right;' >
                        <button id='".$row['AppointmentID']."' class='btn btn-dark edit_data'>Edit</button>
                    </td>";
                }
            ?>
        </tbody>
    </table>
</div>
```

Figure 178 Read Appointment Part at Dr\_appointment.php

The table function were called in the script tag. After that the table tag were created and the table header is set right align. For the table data part, the data will be received from the database. The SQL is to Select all appointment info based on the patient ID. Next a while loop will be used to echo all the data found, besides data two button will also be created at the end of each data which is the ‘View’ button and ‘Edit’ button

## Read Treatment Record Notes



The screenshot shows a portion of a script running on a mobile device. The script is written in JavaScript and uses jQuery. It contains logic to handle a 'click' event on elements with the class '.view\_data'. When such an event occurs, it retrieves the 'id' attribute from the clicked element and stores it in a variable 'aid'. If 'aid' is not an empty string, it performs an AJAX request to 'trecdata.php' using 'POST' method, sending 'aid' as data. The success callback of the AJAX call updates the '#view\_detail' element with the received data and opens a modal for '#viewinfo'.

```
<script>
$(document).ready(function(){

    $(document).on('click','.view_data',function(){
        var aid=$(this).attr('id');
        if(aid!=''){
            $.ajax({
                url:"trecdata.php",
                method:'POST',
                data:{tdata01:aid},
                success: function(data){
                    $("#view_detail").html(data);
                    $('#viewinfo').modal('show');
                }
            });
        }
    })
<script>
```

Figure 179 Read Treatment Record Part at Dr\_trec\_rec.php

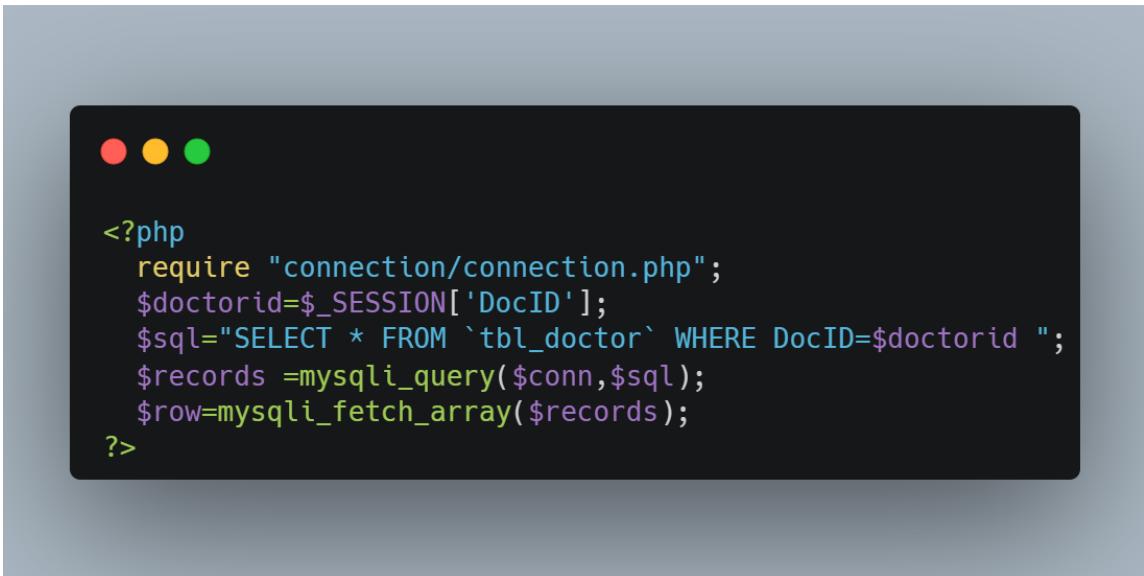
This is the script that the function will call when the user click on the view button on the treatment record page. This function will first save the id of the attribute and put it on a variable call ‘aid’ which stands for appointment ID. After that if aid is not empty string, it will lead to the trecdata.php file.

```
<?php
require 'connection/connection.php';
if(isset($_POST["tdata01"]))
{
    $output = '';
    $query = "SELECT * FROM appointment WHERE AppointmentID = '".$_POST["tdata01"]."'";
    $result = mysqli_query($conn, $query);
    $output .= ' <div class="table-responsive">
        <table class="table table-bordered">';
    while($row = mysqli_fetch_array($result))
    {
        $output .= '
            <tr>
                <td width="30%"><label>AppointmentID</label></td>
                <td width="70%">' . $row["AppointmentID"] . '</td>
            </tr>
            <tr>
                <td width="30%"><label>Time</label></td>
                <td width="70%">' . $row["Time"] . '</td>
            </tr>
            <tr>
                <td width="30%"><label>Date</label></td>
                <td width="70%">' . $row["Date"] . '</td>
            </tr>
            <tr>
                <td width="30%"><label>Notes</label></td>
                <td width="70%">' . $row["Notes"] . '</td>
            </tr>
        ';
    }
    $output .= '
        </table>
    </div>
';
    echo $output;
}
?>
```

Figure 180 Read Treatment Record Part at Dr\_fetch\_view.php

This is the code at trecdata.php, in here it will connect to the database server and having a SQL to select the in the appointment table, acquiring Appointment ID, Time, Date and Notes. After that a while loop will store all the table data get from the SQL into a variable call ‘output’. Lastly, echo output will return the value of output back to the ajax function.

## Read Doctor Account Info

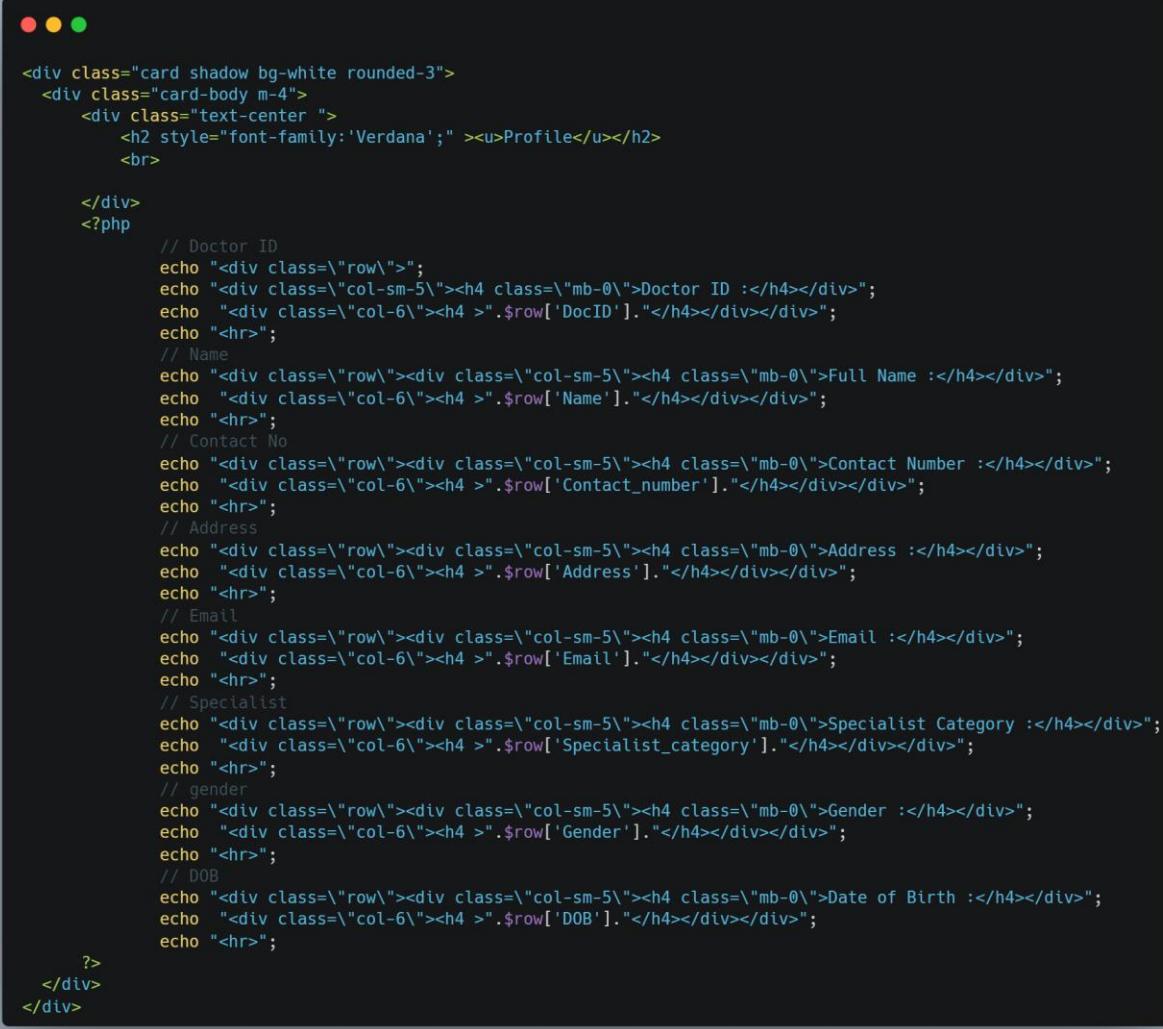


The screenshot shows a terminal window with three colored status indicators (red, yellow, green) at the top. The main area contains the following PHP code:

```
<?php
require "connection/connection.php";
$doctorid=$_SESSION['DocID'];
$sql="SELECT * FROM `tbl_doctor` WHERE DocID=$doctorid ";
$records =mysqli_query($conn,$sql);
$row=mysqli_fetch_array($records);
?>
```

Figure 181 Read Doctor Account part at Dr\_acc.php

In this part, the code connects to the database and have a SQL of select all data from the doctor table where the Doctor ID equal to the session Doctor ID. After that, saving all those data in the ‘records’ variable



```
<div class="card shadow bg-white rounded-3">
<div class="card-body m-4">
    <div class="text-center ">
        <h2 style="font-family:'Verdana';" ><u>Profile</u></h2>
        <br>
    </div>
<?php
    // Doctor ID
    echo "<div class=\"row\">";
    echo "<div class=\"col-sm-5\"><h4 class=\"mb-0\">Doctor ID :</h4></div>";
    echo " <div class=\"col-6\"><h4 >".$row['DocID']. "</h4></div></div>";
    echo "<hr>";
    // Name
    echo "<div class=\"row\"><div class=\"col-sm-5\"><h4 class=\"mb-0\">Full Name :</h4></div>";
    echo " <div class=\"col-6\"><h4 >".$row['Name']. "</h4></div></div>";
    echo "<hr>";
    // Contact No
    echo "<div class=\"row\"><div class=\"col-sm-5\"><h4 class=\"mb-0\">Contact Number :</h4></div>";
    echo " <div class=\"col-6\"><h4 >".$row['Contact_number']. "</h4></div></div>";
    echo "<hr>";
    // Address
    echo "<div class=\"row\"><div class=\"col-sm-5\"><h4 class=\"mb-0\">Address :</h4></div>";
    echo " <div class=\"col-6\"><h4 >".$row['Address']. "</h4></div></div>";
    echo "<hr>";
    // Email
    echo "<div class=\"row\"><div class=\"col-sm-5\"><h4 class=\"mb-0\">Email :</h4></div>";
    echo " <div class=\"col-6\"><h4 >".$row['Email']. "</h4></div></div>";
    echo "<hr>";
    // Specialist
    echo "<div class=\"row\"><div class=\"col-sm-5\"><h4 class=\"mb-0\">Specialist Category :</h4></div>";
    echo " <div class=\"col-6\"><h4 >".$row['Specialist_category']. "</h4></div></div>";
    echo "<hr>";
    // gender
    echo "<div class=\"row\"><div class=\"col-sm-5\"><h4 class=\"mb-0\">Gender :</h4></div>";
    echo " <div class=\"col-6\"><h4 >".$row['Gender']. "</h4></div></div>";
    echo "<hr>";
    // DOB
    echo "<div class=\"row\"><div class=\"col-sm-5\"><h4 class=\"mb-0\">Date of Birth :</h4></div>";
    echo " <div class=\"col-6\"><h4 >".$row['DOB']. "</h4></div></div>";
    echo "<hr>";
?
</div>
</div>
```

Figure 182 Read Doctor Account part at Dr\_acc.php

In this part the data read earlier are print in a card body.

### 18.8.3 Update – HTML Form, PHP

#### Update Treatment Record



```
<script>
$(document).on('click','.edit_data',function(){
    var aid=$(this).attr('id');
    $.ajax({
        url:"fetch.php",
        method:'POST',
        data:{aid:aid},
        dataType:"json",
        success: function(data){
            $("#notes").val(data.Notes);
            $("#appid").val(data.AppointmentID);
            $("#insert").val("Update");
            $('#editinfo').modal('show');
        }
    });
});

$('#insert_form').on("submit",function(event){
    event.preventDefault();
    if($('#notes').val()==''){
        alert("Notes is required")
    }
    else{
        $.ajax({
            url:"insert.php",
            method:"POST",
            data:$('#insert_form').serialize(),
            success:function(data){
                $('#editinfo').modal('hide');
                alert("Record updated Successfully");
            }
        })
    }
})
</script>
```

Figure 183 Update Treatment Record at Dr\_treat-rec02.php

To update treatment record, the process of the user needs to made is to click on the edit button and fill in the form show and then click on the submit button. The code above is showing the two functions of the two buttons.

The edit button will start a ajax function and lead to the “Dr\_fetch\_edit.php” file.

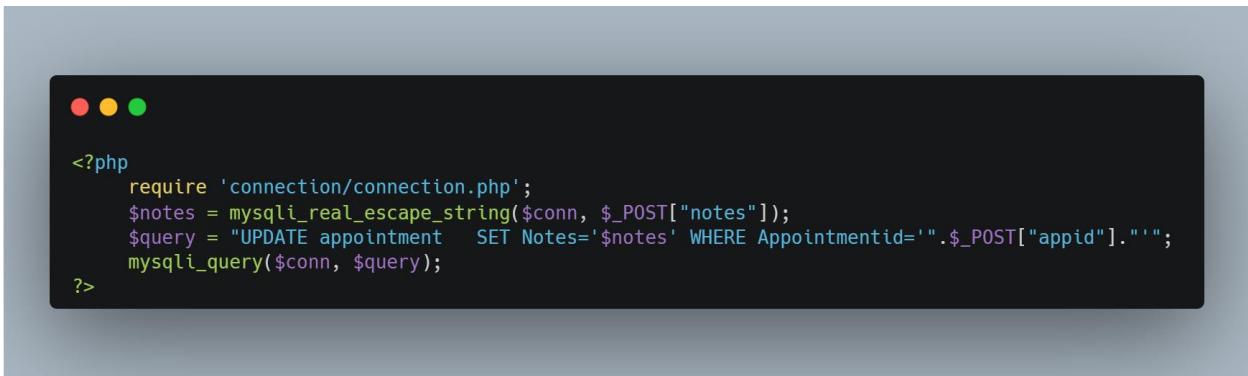


```
<?php
require 'connection/connection.php';
if(isset($_POST["aid"]))
{
    $output = '';
    $query = "SELECT * FROM appointment WHERE AppointmentID = '".$_POST["aid"]."'";
    $result01 = mysqli_query($conn, $query);
    $row=mysqli_fetch_array($result01);
    echo json_encode($row);
}
```

Figure 184 Update Treatment Record part at Dr\_fetch\_edit.php

This file will connect to the database and select all of the data from appointment table and then return the result back to the ajax function. The ajax function will then echo the data in respective places. For example, the “notes” data will be insert into the value of the form entry box.

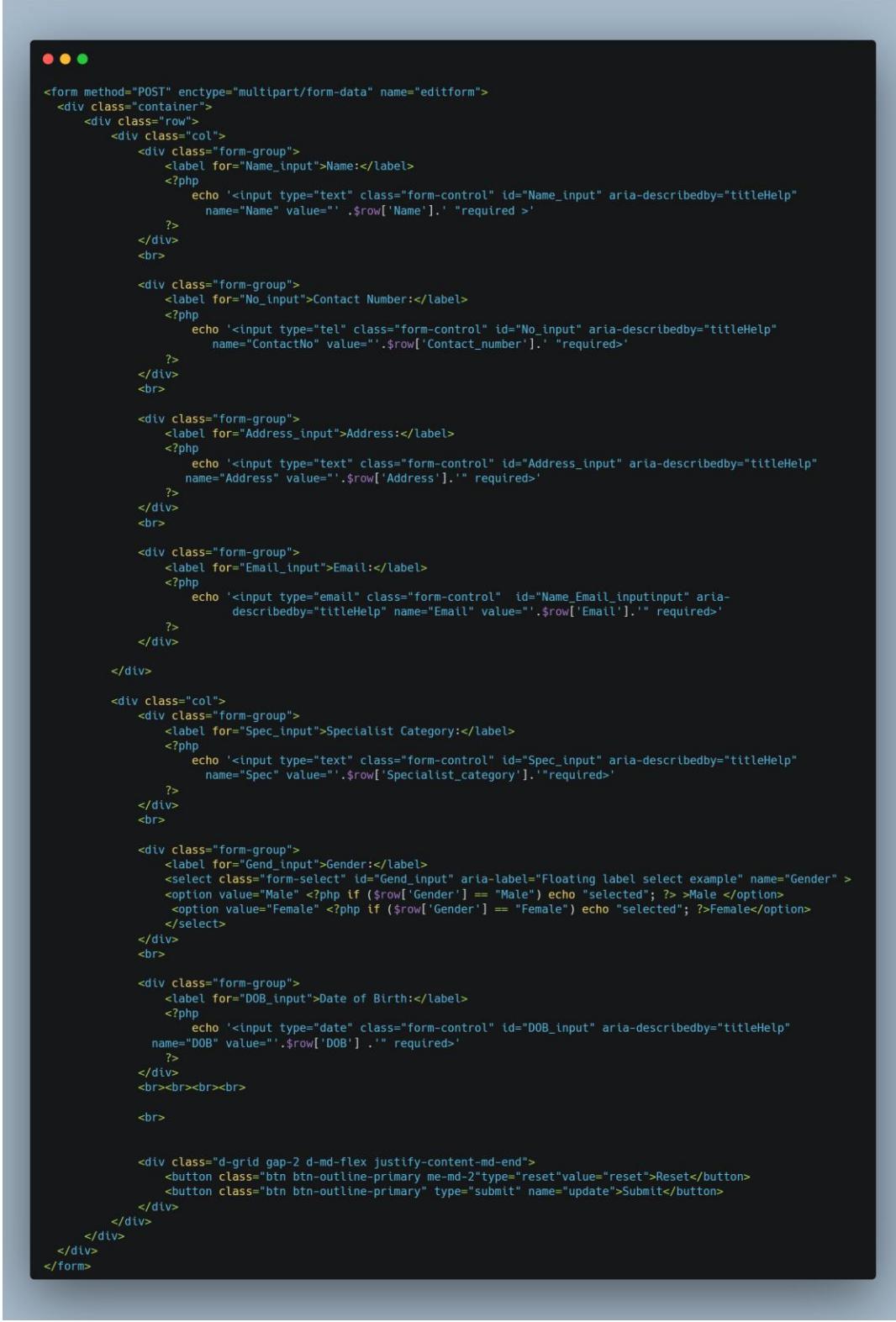
After the user made changes and wanted to update the data in database, they will click on the submit button. The submit button will start another ajax function which lead to the “Dr\_update\_trec.php”.



```
<?php
require 'connection/connection.php';
$notes = mysqli_real_escape_string($conn, $_POST["notes"]);
$query = "UPDATE appointment SET Notes='".$notes."' WHERE Appointmentid='".$_POST["appid"]."'";
mysqli_query($conn, $query);
?>
```

This function will connect to the database first, and then having an update function setting the new data replacing all of the previous data.

## Update Profile Information



```
<?php  
$row = $result->fetch_assoc();  
  
<form method="POST" enctype="multipart/form-data" name="editform">  
    <div class="container">  
        <div class="row">  
            <div class="col">  
                <div class="form-group">  
                    <label for="Name_input">Name:</label>  
                    <?php  
                        echo '<input type="text" class="form-control" id="Name_input" aria-describedby="titleHelp" name="Name" value="'.$row['Name'].'" required>'  
                    ?>  
                </div>  
                <br>  
  
                <div class="form-group">  
                    <label for="No_input">Contact Number:</label>  
                    <?php  
                        echo '<input type="tel" class="form-control" id="No_input" aria-describedby="titleHelp" name="ContactNo" value="'.$row['Contact_number'].'" required>'  
                    ?>  
                </div>  
                <br>  
  
                <div class="form-group">  
                    <label for="Address_input">Address:</label>  
                    <?php  
                        echo '<input type="text" class="form-control" id="Address_input" aria-describedby="titleHelp" name="Address" value="'.$row['Address'].'" required>'  
                    ?>  
                </div>  
                <br>  
  
                <div class="form-group">  
                    <label for="Email_input">Email:</label>  
                    <?php  
                        echo '<input type="email" class="form-control" id="Email_input" aria-describedby="titleHelp" name="Email" value="'.$row['Email'].'" required>'  
                    ?>  
                </div>  
  
            </div>  
            <div class="col">  
                <div class="form-group">  
                    <label for="Spec_input">Specialist Category:</label>  
                    <?php  
                        echo '<input type="text" class="form-control" id="Spec_input" aria-describedby="titleHelp" name="Spec" value="'.$row['Specialist_category'].'" required>'  
                    ?>  
                </div>  
                <br>  
  
                <div class="form-group">  
                    <label for="Gend_input">Gender:</label>  
                    <select class="form-select" id="Gend_input" aria-label="Floating label select example" name="Gender" >  
                        <option value="Male" ><?php if ($row['Gender'] == "Male") echo "selected"; ?> Male </option>  
                        <option value="Female" ><?php if ($row['Gender'] == "Female") echo "selected"; ?> Female </option>  
                    </select>  
                </div>  
                <br>  
  
                <div class="form-group">  
                    <label for="DOB_input">Date of Birth:</label>  
                    <?php  
                        echo '<input type="date" class="form-control" id="DOB_input" aria-describedby="titleHelp" name="DOB" value="'.$row['DOB'].'" required>'  
                    ?>  
                </div>  
                <br><br><br><br>  
            <br>  
  
            <div class="d-grid gap-2 d-md-flex justify-content-md-end">  
                <button class="btn btn-outline-primary me-md-2" type="reset" value="reset">Reset</button>  
                <button class="btn btn-outline-primary" type="submit" name="update">Submit</button>  
            </div>  
        </div>  
    </div>  
</form>
```

Figure 185 Update Profile Information part at Dr\_editprofile.php

The code in figure 124 is the update form which include original data value read from the database. There are two button after the form which is the ‘reset’ button and ‘submit’ button. The ‘reset’ button is use to reset all value in the entry box back to its original value, however the submit button is submit all new value.



```
<?php
if(isset($_POST['update']))
{
    require "connection/connection.php";
    $doctorid=$_SESSION['DocID'];
    $name=$_POST['Name'];
    $ContNo=$_POST['ContactNo'];
    $Add=$_POST['Address'];
    $email=$_POST['Email'];
    $SpecCat=$_POST['Spec'];
    $gender=$_POST['Gender'];
    $DOB=$_POST['DOB'];

    $sql= "UPDATE tbl_doctor SET Name='$name',Contact_number='$ContNo',Address='$Add',Email='$email'
           ,Specialist_category='$SpecCat',Gender='$gender',DOB='$DOB' WHERE DocID=$doctorid ";

    if(!mysqli_query($conn,$sql))
    {
        echo "<script>alert('Failed : Update failed')</script>";
    }else{
        echo "<script>alert('Account has been updated')
        window.location.href = 'Dr_acc.php'
        </script>";
    }
}
?>
```

Figure 186 Update Account Information part in Dr\_editprofile.php

This is the code that update the latest data that collected in the form in figure 124 into the database. A update SQL statement is use to update the data into the doctor table, if an error occur during the update, an alert box will pop up and notice user ‘Failed: Update Failed’, however if the the data had been updated to the database successfully, an alert box will pop up noticing user ‘ Account has been updated’ and back to the Doctor account page.

## Update Profile Picture



```
<form method="POST" enctype="multipart/form-data">
    <div class="form-group">
        <input type="file" accept="image/*" class="form-control" id="Propic" name="Propic" required>
    </div>
    <br>
    <div class="d-grid gap-2 d-md-flex justify-content-md-end">
        <button type="button" class="btn btn-secondary me-md-2" data-bs-dismiss="modal">Cancel</button>
        <button class="btn btn-primary float-end" type="submit" name="update">Upload</button>
    </div>
</form>
```

Figure 187 Update Profile Picture part at Dr\_acc.php

The code above is the form to receive data entry from the user. The input type use is ‘file’ and it had been set that only image file is acceptable. There is two button that is created which is ‘cancel’ button and ‘Upload’ button. The ‘Cancel’ button will do nothing and close the modal, and the ‘Upload’ button will submit the form and save the input value into the database.



```
<?php
if(isset($_POST['update']))
{
    require "connection/connection.php";
    $doctorid=$_SESSION['DocID'];
    $propic=$_FILES['Propic']['tmp_name'];
    $img=addslashes(file_get_contents($propic));
    $sql= "UPDATE tbl_doctor SET Profile_pic='{$img}' WHERE DocID=$doctorid ";

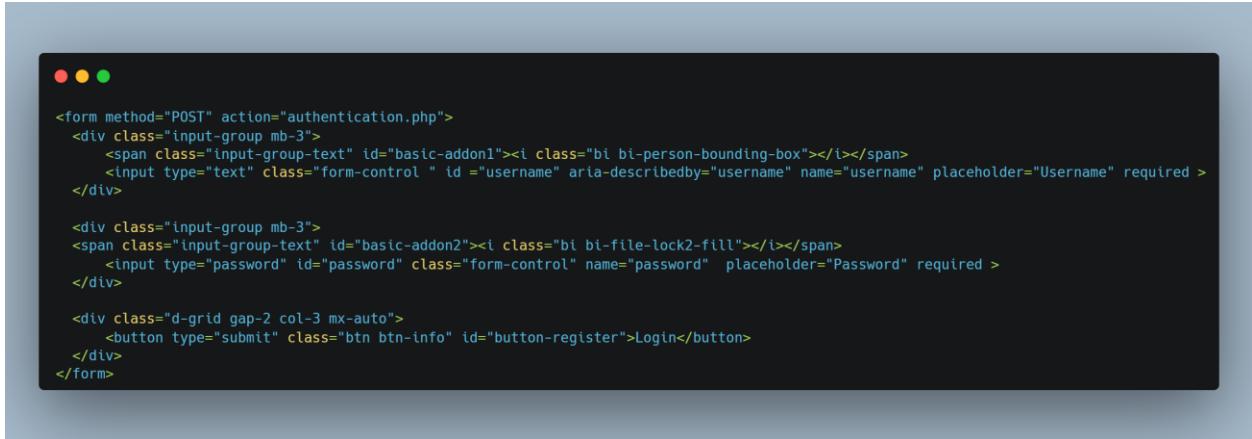
    if(!mysqli_query($conn,$sql))
    {
        echo "<script>alert('Failed : Update failed')</script>";
    }else{
        echo "<script>alert('Profile Picture had been update')
        window.location.href = 'Dr_acc.php'
        </script>";
    }
}
?>
```

Figure 188 Update Profile Picture part at Dr\_acc.php

In figure 127, the code shown is the code that function to Update the database with new input value. If the update process failed, an alert box will appear and notice the user ‘Failed: Update failed’. If the update process succeed, an alert box will appear and notice the user “Profile Picture had been update” and direct the user back to the doctor account page.

#### 18.8.4 Login – HTML Form, PHP

##### Login Form



The figure shows a screenshot of a code editor displaying the HTML code for a login form. The code is written in a monospaced font and includes various CSS classes and Bootstrap grid components. The code starts with a form tag set to POST method and action 'authentication.php'. It contains two input groups, each with a prefix icon (person and lock) and a required input field for username and password respectively. Below the inputs is a button group containing a single 'Login' button.

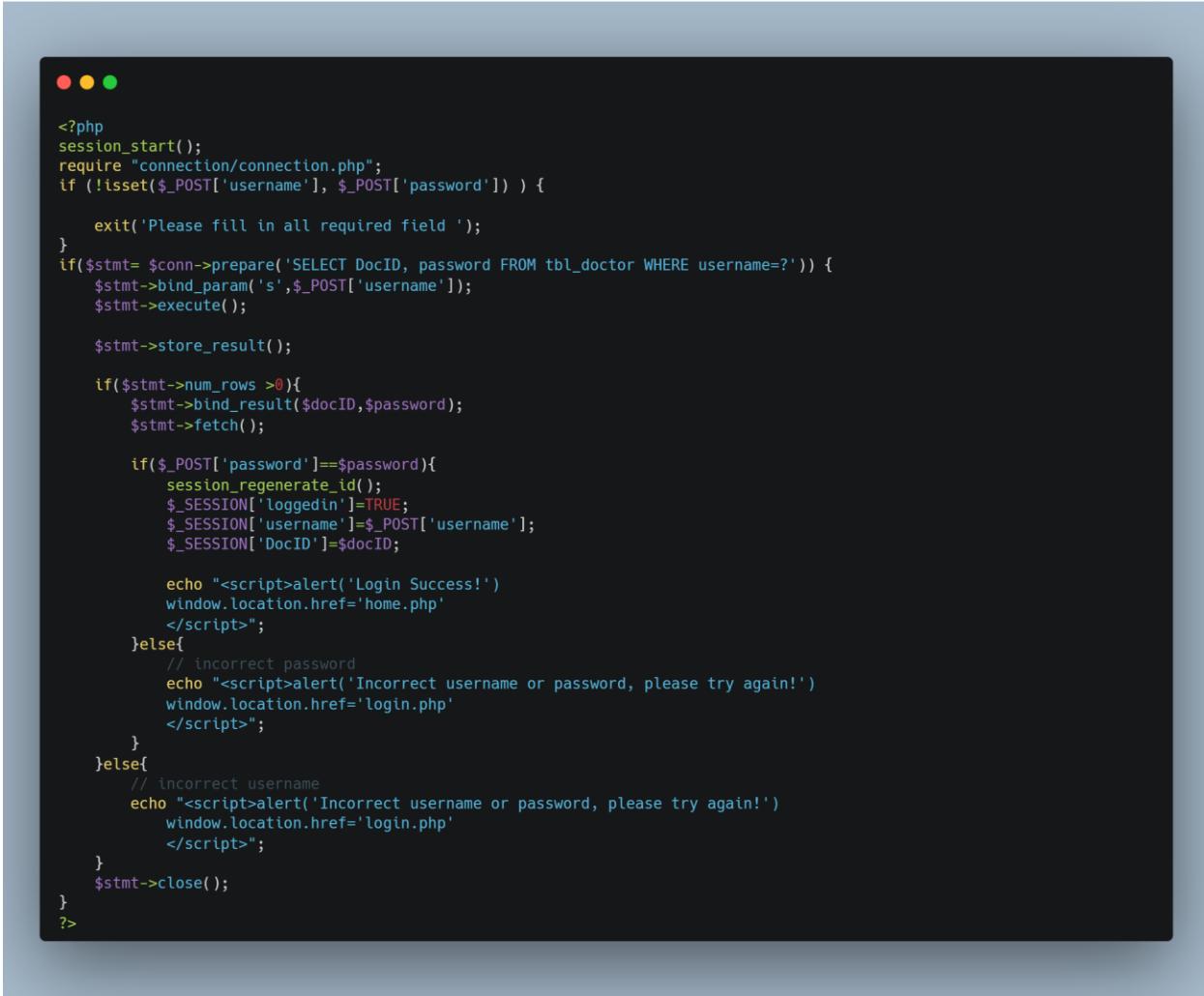
```
<form method="POST" action="authentication.php">
<div class="input-group mb-3">
  <span class="input-group-text" id="basic-addon1"><i class="bi bi-person-bounding-box"></i></span>
  <input type="text" class="form-control" id="username" aria-describedby="username" name="username" placeholder="Username" required>
</div>

<div class="input-group mb-3">
  <span class="input-group-text" id="basic-addon2"><i class="bi bi-file-lock2-fill"></i></span>
  <input type="password" class="form-control" id="password" name="password" placeholder="Password" required>
</div>

<div class="d-grid gap-2 col-3 mx-auto">
  <button type="submit" class="btn btn-info" id="button-register">Login</button>
</div>
</form>
```

Figure 189 Login Form Part at Dr\_login.php

The figure 128 show the code about the login form at the login page. Like normal login, it require user to enter both username and password, therefore there will be two input box which is for username and password entry. A submit button type were also created as the login button.



```
<?php
session_start();
require "connection/connection.php";
if (!isset($_POST['username'], $_POST['password'])) {
    exit('Please fill in all required field ');
}
if($stmt= $conn->prepare('SELECT DocID, password FROM tbl_doctor WHERE username=?')) {
    $stmt->bind_param('s',$_POST['username']);
    $stmt->execute();
    $stmt->store_result();

    if($stmt->num_rows >0){
        $stmt->bind_result($docID,$password);
        $stmt->fetch();

        if($_POST['password']==$password){
            session_regenerate_id();
            $_SESSION['loggedin']=TRUE;
            $_SESSION['username']= $_POST['username'];
            $_SESSION['DocID']=$docID;

            echo "<script>alert('Login Success!')"
            window.location.href='home.php'
            </script>";
        }else{
            // incorrect password
            echo "<script>alert('Incorrect username or password, please try again!')"
            window.location.href='login.php'
            </script>";
        }
    }else{
        // incorrect username
        echo "<script>alert('Incorrect username or password, please try again!')"
        window.location.href='login.php'
        </script>";
    }
    $stmt->close();
}
?>
```

Figure 190 Login Authentication at Dr\_authentication.php

In figure 129, this is the Dr\_authentication file which it will run after the user click on the ‘login’ button at the code of figure 128. At first, the program connection to the database and then checking whether the entry field of username and password is filled or not. If it is not filled, the system will notice user “Please fill in all required field”. After that a statement will prepare the SQL function of selecting doctor ID and password from the doctor table where the username is unspecified. The statement will also bind a string parameter which is the username value get from the form, the statement will then execute the SQL function. After finishing the execution, the result will be stored, and a if else statement will be appear to validate the login credentials. If the execution of the SQL return any results, the statement will bind the results into two variables which is the \$docID and \$password. After that the password filled earlier in the form will be compare with the \$password. If it is same, multiple session element will be save and the user will be directed to the doctor homepage. If the password is not same the user will be notice that incorrect password or username and direct to back to the login page. If there are no username found, the user will also be notice that incorrect password or username and direct back to the login page again.

## 18.9 Staff Module

### 18.9.1 Create – HTML Form, PHP

#### Create Database

```
CREATE TABLE `tbl_account` (
  `Acc_ID` int(11) NOT NULL,
  `Name` varchar(50) DEFAULT NULL,
  `Phone_number` varchar(50) DEFAULT NULL,
  `Email` varchar(50) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

-- 
-- Dumping data for table `tbl_account`
-- 

INSERT INTO `tbl_account` (`Acc_ID`, `Name`, `Phone_number`, `Email`) VALUES
(1, 'Matty Ornelas', '9347752505', 'mornelas@reddit.com'),
(2, 'Roselle Wolffers', '6458311881', 'rwolffers1@nba.com'),
(3, 'Shayne Fetherston', '9412384310', 'sfetherston2@kickstarter.com'),
(4, 'Erny Scarrisbrick', '6513010516', 'escarrisbrick3@delicious.com'),
(5, 'Rowan Riccione', '1412925179', 'rriccione4@google.co.uk'),
(6, 'Angelica Lukasen', '9538114431', 'alukasen5@ameblo.jp'),
(7, 'Willard Brydone', '8044809701', 'wbrydone6@spiegel.de'),
(8, 'Woodman Marsland', '9509563038', 'wmarsland7@netscape.com'),
(9, 'Jobyna Braganca', '7185514875', 'jbraganca8@unblog.fr'),
(10, 'Wanids Espinosa', '4505000964', 'wespinosa9@typepad.com'),
(11, 'Reine Spreckley', '7307130699', 'rspreckleya@moonfruit.com'),
(12, 'Adriaens De Brett', '9668090274', 'adeb@elpais.com'),
(13, 'Gussy Eads', '2098434129', 'geadsc@java.com'),
(14, 'Giselbert Bilbey', '3586122100', 'gbilbeyd@theatlantic.com'),
(15, 'Cornall Marvel', '2619158560', 'cmarvelc@yale.edu'),
(16, 'Truda Foddy', '7049095434', 'tfoddyf@apple.com'),
(17, 'Pierson Spears', '5771335216', 'pspearsg@etsy.com'),
(18, 'Suzanne Dellenbrook', '7416224169', 'sdellenbrookh@adobe.com'),
(19, 'Annetta Ducker', '2385488366', 'aduckeri@addtoany.com'),
(20, 'Devinne Rickards', '8716154704', 'drickardsj@addthis.com');
```

Figure 191 Create Staff Database

Based on the picture above, the codes are used to create the tables for the database and the data to be stored into their respective tables. For example, we can see that the statement “CREATE” is used to create tables in the database. And for data to be inserted to the table, the statement “INSERT” is used to insert database into the table.

## 18.9.2 Read – HTML Form, PHP

### Read Username and Password

```
if (empty($username) && empty($password))
{
    header("Location: staff_login.php?error=Fields cannot be empty!");
    exit();
}
else if (empty($username))
{
    header("Location: staff_login.php?error=Username is required!");
    exit();
}
else if(empty($password))
{
    header("Location: staff_login.php?error=Password is required!");
    exit();
}
else
{
    $password = md5($password);
    $sql = "SELECT * FROM `tbl_staff_acc` WHERE Username = '$username' AND Password = '$password'";

    $result = mysqli_query($conn, $sql);

    if (mysqli_num_rows($result) == 1)
    {
        $row = mysqli_fetch_assoc($result);

        if ($row['Username'] === $username && $row['Password'] === $password)
        {
            $_SESSION['Staff_Username'] = $row['Username'];

            header("Location: staff_dashboard.php");
        }
        else
        {
            header("Location: staff_login.php?error= Incorrect Username or Password!");
            exit();
        }
    }
}
```

Figure 192 Read username and password

The username and password will be stored in a variable and then later be used for later parts of the code. On the IF statement where (empty(\$username) && empty(\$password)) means that if the fields of the username and password is empty it will execute the code and the code where “header(“Location: staff\_login.php?error=Fields cannot be empty!&userdata”)” it will display that message on top of the form. As for the rest of the else if statements it will be the same concept as explained. If the staff has entered everything into the form, the else statement will be executed. It will read the data from the form that the user has inputted and then it will verify the data into from database table where the code “if (\$row['Username'] === \$username && \$row['Password'] === \$password)

```
{
    $_SESSION['Staff_Username'] = $row['Username'];

    header("Location: staff_dashboard.php");
```

}

)”.

### 18.9.3 Update – HTML Form, PHP

#### Update Medicine

```
</php>
if(isset($_POST['update']))
{
    $MedID = $_POST['MedID'];

    $sql = "SELECT *
    FROM `tbl_medicine`
    WHERE `MedID` = '$MedID'";

    $result = mysqli_query($conn, $sql);

    while($row = mysqli_fetch_array($result))
    {
        >>>

        <form class="w-5 vh-100" action="update_medicine.php" method="post">
            <input type="hidden" name="MedID" value=</php> echo $row['MedID'] />>>

            <div class="mb-3">
                <label for="title" class="form-label">Name</label>
                <input class="form-control" type="text" placeholder="Title" name="Name" aria-label="Title" value=</php> echo $row['Name'] />>> required
            </div>
            <div class="mb-3">
                <label for="Description" class="form-label">Description</label>
                <textarea class="form-control" placeholder="Description" name="Description" aria-label="Description" style="height: 100px" required></php> echo $row['Description'] />>>
            </div>
            <div class="mb-3">
                <label for="Specialist_category" class="form-label">Price</label>
                <input class="form-control" type="text" placeholder="Price" name="Price" aria-label="Specialist Category" value=</php> echo $row['Price'] />>> required
            </div>
            <div class="mb-3">
                <label for="Consultation_Fees" class="form-label">Quantity</label>
                <input class="form-control" type="text" placeholder="Quantity" name="Quantity" value=</php> echo $row['Quantity'] />>> required
            </div>
            <div class="d-grid">
                <button type="submit" class="btn btn-dark w-5" name="update">Submit</button>
            </div>
        </form>
    </div>
}

</php>
include("conn.php");

if(isset($_POST['update']))
{
    function validate($data)
    {
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
    }

    $title = validate($_POST['Name']);
    $description = validate($_POST['Description']);
    $specialist_category = validate($_POST['Price']);
    $consultation_fees = validate($_POST['Quantity']);

    $check_existing = mysqli_query($conn, "SELECT * FROM `tbl_medicine` WHERE MedID = '".$_POST[MedID]'");
    $exist_rows = mysqli_num_rows($check_existing);

    if (!$exist_rows == 1)
    {
        header("Location: update_medicine_form.php?error= Please Try Again !");
    }
    else
    {
        $sql = "UPDATE `tbl_medicine` SET
        'Name' = '$title',
        'Description' = '$description',
        'Price' = '$specialist_category',
        'Quantity' = '$consultation_fees'
        WHERE `MedID` = '".$_POST[MedID]+"'";

        if(mysqli_query($conn,$sql))
        {
            header("Location: update_medicine_form.php?success= Medicine Successfully Updated !");
        }
        else
        {
            header("Location: update_medicine_form.php?error= Please Try Again !");
        }
    }
}

mysqli_close($conn);
>>
```

*Figure 193 Update Medicine*

Based on the first picture, the medicine data will be displayed in the form and will get the data from the databased based on the staff session and MedID that the staff has click. If the user decides to make changes to the medicine, for example the price or name, they will need to backspace the current data displayed on the form and then enter the new credentials. Once entered the staff will need to click on the “Update” button. The form will then execute the codes based on the second picture as shown, the data will be stored in the variables and then the code “\$sql=”UPDATE users SET Name = ‘\$title’, Description = '\$description, Price = '\$specialist\_category', Quantity = '\$consultation\_fees' WHERE MedID = ‘\$\_POST[MedID]’” it will replace the old data with the new data that the staff has keyed in with the update function.

## 18.9.4 Login – HTML Form, PHP

### Staff Login

```
<!-- -->
<div class="row justify-content-between align-items-center mb-5">
  <div class="col">
    <h3 class="card-title text-center">Login</h3>
  </div>
</div>

<php if (isset($_GET['error']))
{
  echo '<div class="alert alert-danger m-5 text-center" role="alert">'.
        $_GET['error'].
  '</div>';
} >

<form class="mx-5" action="staff_logincheck.php" method="post">
  <div class="mb-3">
    <label for="Username" class="form-label">Username</label>
    <input class="form-control" type="text" placeholder="Username" name="Username" aria-label="Username">
  </div>
  <div class="mb-3">
    <label for="Password" class="form-label">Password</label>
    <input type="password" class="form-control" placeholder="Password" name="Password">
  </div>
  <div class="d-grid">
    <button type="submit" class="btn btn-dark mt-5 mb-5 rounded">Submit</button>
  </div>
</form>
<div class="text-danger text-center">
  <h6>If you are not registered, kindly approach towards admins for registration. </h6>
</div>

if (empty($username) && empty($password))
{
  header("Location: staff_login.php?error=Fields cannot be empty!");
  exit();
}
else if (empty($username))
{
  header("Location: staff_login.php?error=Username is required!");
  exit();
}
else if(empty($password))
{
  header("Location: staff_login.php?error=Password is required!");
  exit();
}
else
{
  $password = md5($password);
  $sql = "SELECT * FROM `tbl_staff_acc` WHERE Username = '$username' AND Password = '$password'";

  $result = mysqli_query($conn, $sql);

  if (mysqli_num_rows($result) == 1)
  {
    $row = mysqli_fetch_assoc($result);

    if ($row['Username'] === $username && $row['Password'] === $password)
    {
      $_SESSION['Staff_Username'] = $row['Username'];

      header("Location: staff_dashboard.php");
    }
    else
    {
      header("Location: staff_login.php?error= Incorrect Username or Password!");
      exit();
    }
  }
}
```

Figure 194 Staff Login

Based on the first picture, when the staff clicks the sign-in button on the form it will execute the codes based on the second picture. The username and password will be stored in a variable and then later be used for later parts of the code. On the IF statement where (empty(\$username) &&

`empty($password))` means that if the fields of the username and password is empty it will execute the code and the code where “`header("Location: staff_login.php?error=Fields cannot be empty!&userdata")`” it will display that message on top of the form. As for the rest of the else if statements it will be the same concept as explained. If the staff has entered everything into the form, the else statement will be executed. It will read the data from the form that the user has inputted and then it will verify the data into from database table where the code “`if ($row['Username'] === $username && $row['Password'] === $password)`

```
{  
    $_SESSION['Staff_Username'] = $row['Username'];  
    header("Location: staff_dashboard.php");  
}
```

”). Once the code has been executed, it will redirect the staff to the homepage.

## **19.0 Conclusion**

### **19.1 Assumptions**

To start of the journal of our customer in our web system, the user are expected to conceive numerous number of specifications which although might sounds complicated yet simple as the users who are from younger generation would simply posses these skills naturally. The specifications which were mentioned are of the technical skills such as computer skills which are a must have in order to operate the system. Although, the users are not required to have extensive knowledge on how the system operates, the fundamental knowledge such as the process or step-by-step needed to perform certain functions are required in order to fully immerse with our web system. For example, the user are required to posses computer skills which would help them immensely during the booking process as well as navigating through the web system in general. Aside from that, more on the technical requirements are that, users are required to be in possession of an electronic devices as well as a stable Internet connection in order to access our web system. Furthermore, users are also assumed to only access certain features such as to make bookings which is also accessible for registered users, therefore, any guest who wish to utilize such feature are required to register themselves prior to making a booking.

For the customer, all users are assumed to have an account in order to make an appointment with the doctor for treatment or consultation purposes. Registered customers should be able to choose their payment method upon check-out either credit card or cash. Once payment has been received by the clinic, the assumption is that the customer should be able to view their appointment date and time at their booking history.

In the admin dashboard of Valley Clinic management system, all type of users is assumed to create account with the assistance of an Admin. This will increase the security of the entire system as anonymous user are unable to register account is the management system. While creating account for admin, doctor, and staff, the create account form should be filled up by the account holder. The doctor account will update his or her information in the doctor dashboard in future. Update function in admin dashboard is only allowed for treatment and services. This will avoid admin from simply editing personal information of other doctors, admins, and staffs. Lastly, there will also be an assumption that admin can remove doctor, admin, and staff account. Admin cannot

remove the account as desired but can only remove the account whenever requested. The staff, admin, or doctor may leave the clinic, so the human resources of the clinic can request admin to remove the account from the clinic management system.

The doctor account is assumed to be created by the admin. And the doctor will be able to change their account information by themselves after logging in to the account. The admin will have no privilege to edit anything of the doctor account while they can only create or delete the doctor account.

To log into the staff dashboard, all staffs of the clinic are assumed to have their own username and password. After logging in, the staff can add new medicine or edit the medicine information. The assumption of this is that the staffs can make changes which then they can view it under the search function of the system. On the other hand they can remove any medicine that is not needed and the assumption for this is that the medicine that was removed would not be in the system.

## 19.2 Limitations

The limitations for the customer section are not able to cancel their appointments, unable to have their own profile pictures for their respective accounts. This is because, cancellations for the appointments requires refunding the total amount to the customer is a complicated process. As for profile pictures, the system does not have a feature where the registered customer can have their own profile pictures uploaded. This is because, the system is only designed to allow customers to make appointments with the doctors and view what is provided from the clinic.

One of the most significant drawback that could be seen throughout the web system would be the lack of filtering for most majority of the search function. This could be seen in the treatment and services page where the users are only limited to searching based on keyword associated with the treatment or medical condition's name rather than having other filters such as ordering based on price, filter by doctors and etc. Aside from that, the users are also not allowed to access the payment page after the booking process where the user could not pay after the booking session are over.

There are some limitations in the Admin Dashboard are the Admin Dashboard are not able to show the top treatment and services. This is due to the database structure. Our database structure does not connect the payment database and treatment and services database together. Due to this issue, Top treatment and services from the database cannot be retrieved. Furthermore, throughout the navigation of the admin dashboard, admin dashboard is lack of the “back” button which admins needs to click on the navigation icon on the left panel to go to the previous page not clicking the back button to return to the previous page.

The limitations of the doctor part are that for the table in it, the doctors are unable to filter it by category. They can only search for a specific result by entering the keyword in the search bar. Therefore, if they could not filter the data based on date week, time or name.

One of the limitations from the staff dashboard is that the staff could not delete the number of quantities of medicine that is not needed instead it deletes the whole medicine from the database. This is the cause from the medicine table from the database. On the other hand, another limitation is on the appointment function where the staffs are only able to view the appointments for the said

day. Due to this being their only function, they would not be able to make any changes to the appointment even if the clients would like to cancel their appointment. Lastly, the final limitation is there is no back button. This can be seen when the staffs would like to go to the previous page instead on clicking on back they would need to click on the function tab to navigate themselves in the system.

### **19.3 Enhancement**

For future enhancements, the developer will implement the feature where the customer will be able to cancel their appointments if the customer is unable to attend due to unforeseen circumstances and implement uploading profile pictures for respective accounts.

The enhancement that we can do on the Admin Dashboard is to increase more navigation buttons which would allow admin to navigate through pages more easily. Implementing breadcrumb should be a good idea for the admin dashboard. With this implemented, admin will have better interactions with the system.

The enhancement to be done to the doctor part is that to add a dropdown filter button on the table so that the doctors can filter the data with it. This will allow the doctor to do their job more efficiently as they do not need to search the result one by one.

To enhance the system more, the developer would likely allow the staff to have the edit and delete appointment as not all clients are able to cancel or change information regarding their appointments. With this feature, the staffs are able to priority all the patients that are booked for that day.

## 20.0 References

- Anuar, M. A. (2006). Clinic Management System. *Fakulti Sains Komputer dan Teknologi Maklumat*.
- *Bootstrap Icons*. (n.d.). Retrieved from Bootstrap: <https://icons.getbootstrap.com/>
- Bootstrap. (n.d.). *Introduction*. Retrieved from Bootstrap: <https://getbootstrap.com/docs/5.1/getting-started/introduction/>
- Chart.js. (12 December, 2022). *Getting Started*. Retrieved from Chart.js: <https://www.chartjs.org/docs/latest/getting-started/>
- Duke, R. (20 Jan, 2022). *What is a Gantt Chart?* Retrieved from Gantt.com: <https://www.gantt.com/>
- *Entity Relationship Diagram*. (n.d.). Retrieved from SmartDraw: <https://www.smartdraw.com/entity-relationship-diagram/>
- Hamel, G. (27 July, 2018). *Advantages & Disadvantages of Traditional File Organization*. Retrieved from CHRON.: <https://smallbusiness.chron.com/document-management-system-requirements-2881.html>
- *Hierarchy Charts*. (n.d.). Retrieved from MyDraw: <https://www.mydraw.com/diagrams-hierarchy-chart-index>
- *How to Open a Bootstrap Modal Window Using jQuery*. (n.d.). Retrieved from TutorialRepublic: <https://www.tutorialrepublic.com/faq/how-to-open-a-bootstrap-modal-window-using-jquery.php>
- Lutkevich, B. (n.d.). *project scope*. Retrieved from TechTarget: <https://www.techtarget.com/searchcio/definition/project-scope>
- Martins, J. (22 January, 2022). *The quick guide to defining project scope—in 8 steps*. Retrieved from Asana: <https://asana.com/resources/project-scope>
- MKSO75. (28 October, 2020). *Levels in Data Flow Diagrams (DFD)*. Retrieved from GeeksforGeeks: <https://www.geeksforgeeks.org/levels-in-data-flow-diagrams-dfd/>
- *PHP Ajax Update MySQL Data Through Bootstrap Modal*. (n.d.). Retrieved from WebLesson: <https://www.webslesson.info/2016/10/php-ajax-update-mysql-data-through-bootstrap-modal.html>

- *Ultimate Entity Relationship Diagram Tutorial (ER Diagrams)*. (22 February, 2022). Retrieved from Creately: <https://creately.com/blog/diagrams/er-diagrams-tutorial/>

## 21.0 Appendix

Components	Details
Project Objectives	**Discussed in section 6 subtopic 6.4 (Project Aim and Objectives)
Deliverables	<p>1. Product:</p> <p><b><u>Customer Dashboard</u></b></p> <ul style="list-style-type: none"> <li>-Login &amp; registration</li> <li>-booking and payment</li> <li>-view doctors, company, and treatment &amp; services information</li> </ul> <p><b><u>Admin Dashboard</u></b></p> <ul style="list-style-type: none"> <li>-manage treatment and services</li> <li>-manage user account</li> <li>-report</li> </ul> <p><b><u>Doctor Dashboard</u></b></p> <ul style="list-style-type: none"> <li>-manage appointments</li> <li>-view schedule</li> <li>-manage treatment records</li> </ul> <p><b><u>Staff Dashboard</u></b></p> <ul style="list-style-type: none"> <li>-manage medicine</li> <li>-check appointment</li> </ul> <p>2. User Manual</p> <p>3. Instruction and Installation Manual</p>
Milestones	** Discussed in section 7 subtopic 7.2 (Project Gantt Chart)
Technical Requirements	WampServer or XAMP Local Server
Limits / Constraints	** Cover in section 19 subtopic 19.1 and 19.2 (assumptions and limitations)
Necessary Changes	<p>Product:</p> <p><b><u>Customer Dashboard</u></b></p> <ul style="list-style-type: none"> <li>-Add Sort treatment and services with category function</li> </ul>

	<p><b><u>Admin Dashboard</u></b></p> <ul style="list-style-type: none"> <li>- Add top treatment and services in report section</li> </ul> <p><b><u>Doctor Dashboard</u></b></p> <ul style="list-style-type: none"> <li>-Add register leave function</li> </ul> <p><b><u>Staff Dashboard</u></b></p> <ul style="list-style-type: none"> <li>-Add manage appointment function</li> </ul>
--	--

## 21.1 Installation Guide for WampServer



*Figure 150: WampServer main page*

As shown above it the website to download WampServer onto one's desktop or laptop.



*Figure 151: Download WampServer*

Download WampServer setup from the official website. There are two available version to download from. Users can download any version to match their computer's configuration setting.

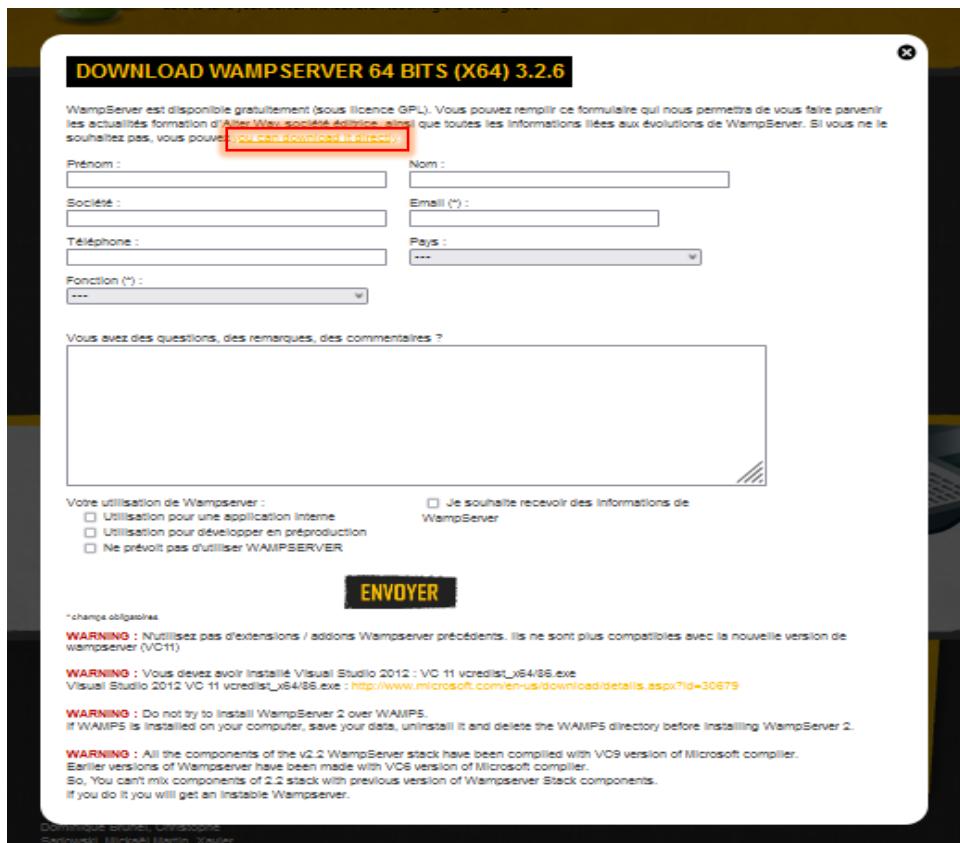


Figure 152: Popup message

This popup message will be shown after clicking on the download button. To continue downloading click on the you can download it directly as highlighted above and continue the download process.

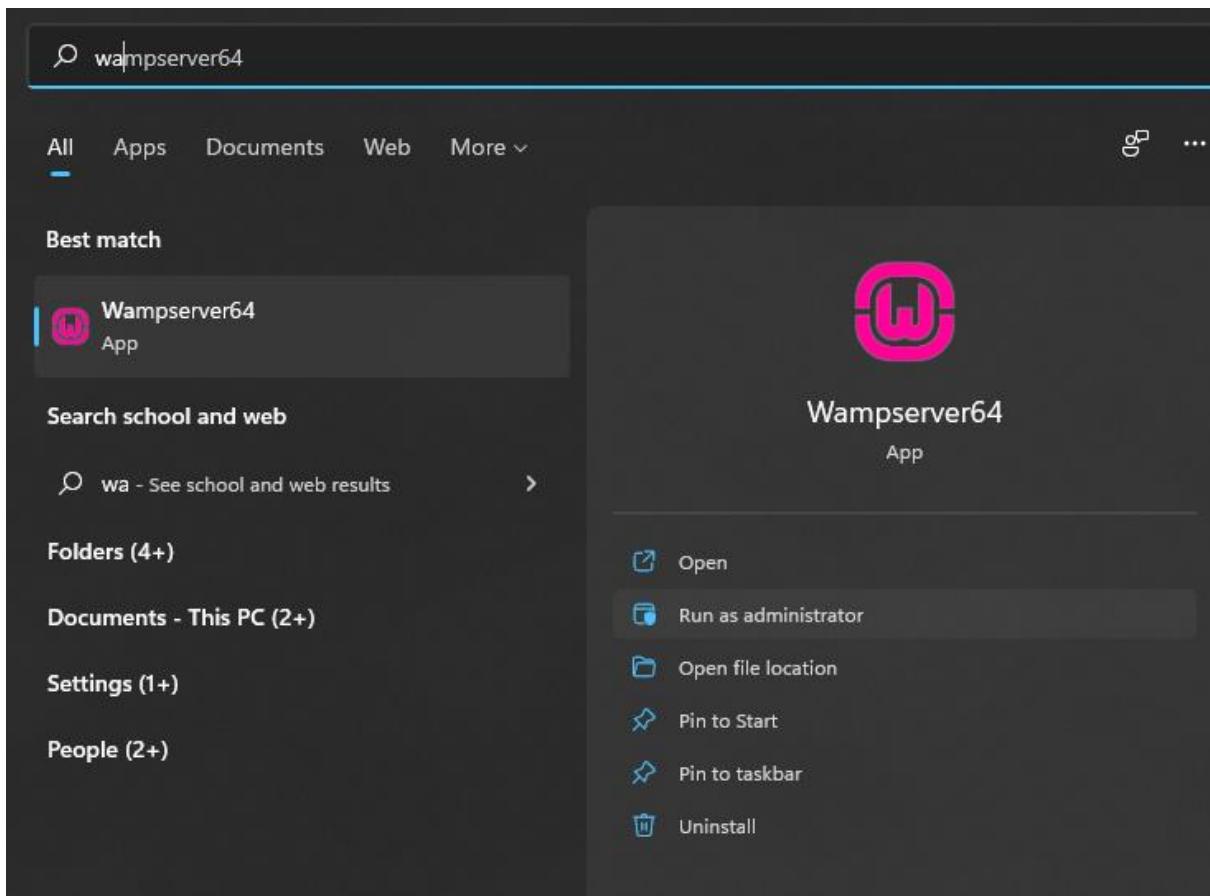


Figure 153: Search WampServer

Search for WampServer to initial the installation process by running it as administrator.

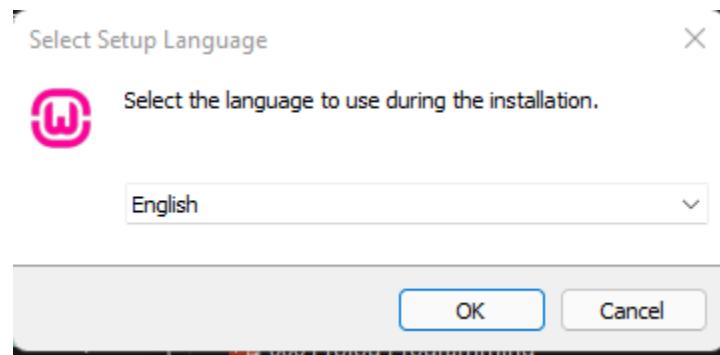
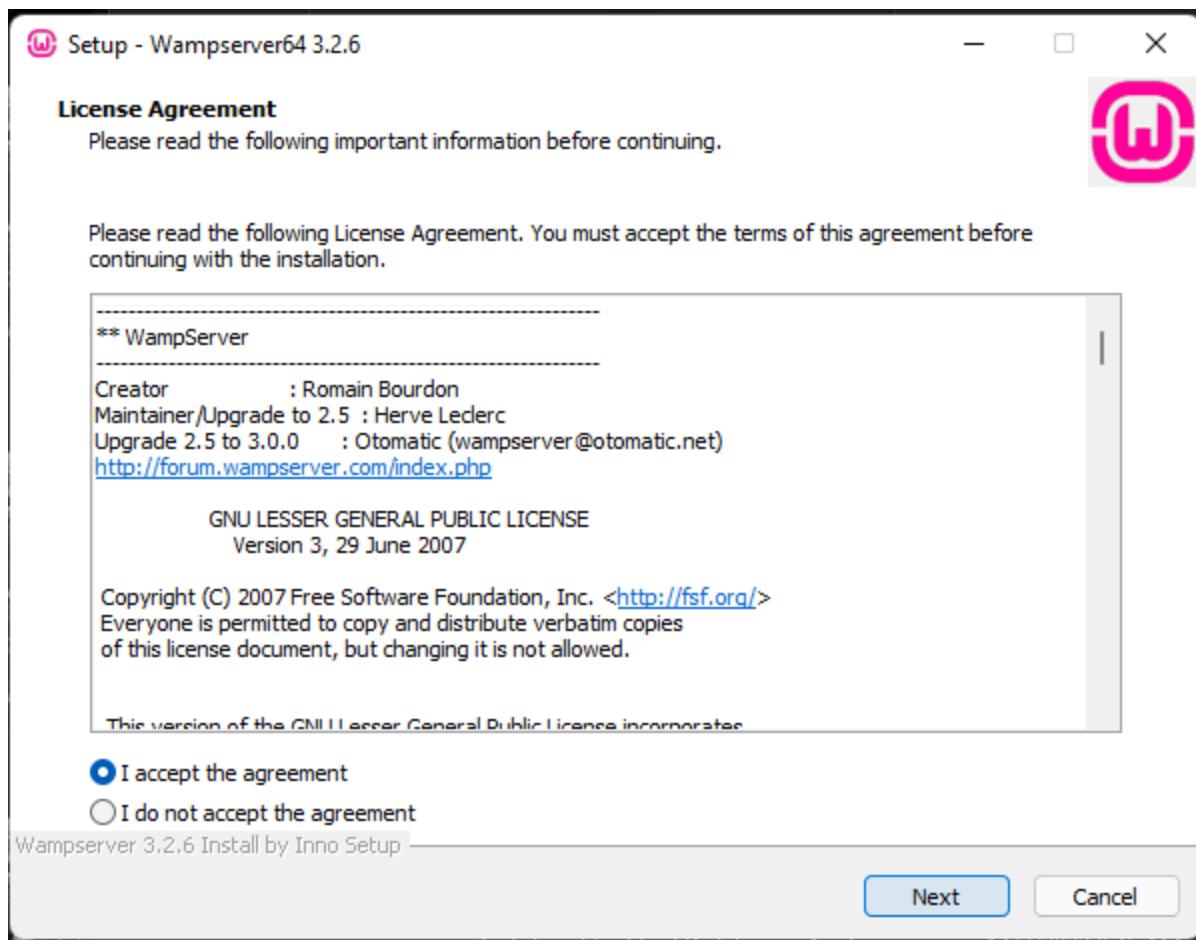


Figure 154: Select Setup Language

Select the preferred language and click on ok.



*Figure 155: License Agreement*

Click on I accept the agreement after reading it to continue the installation process and proceed to next.

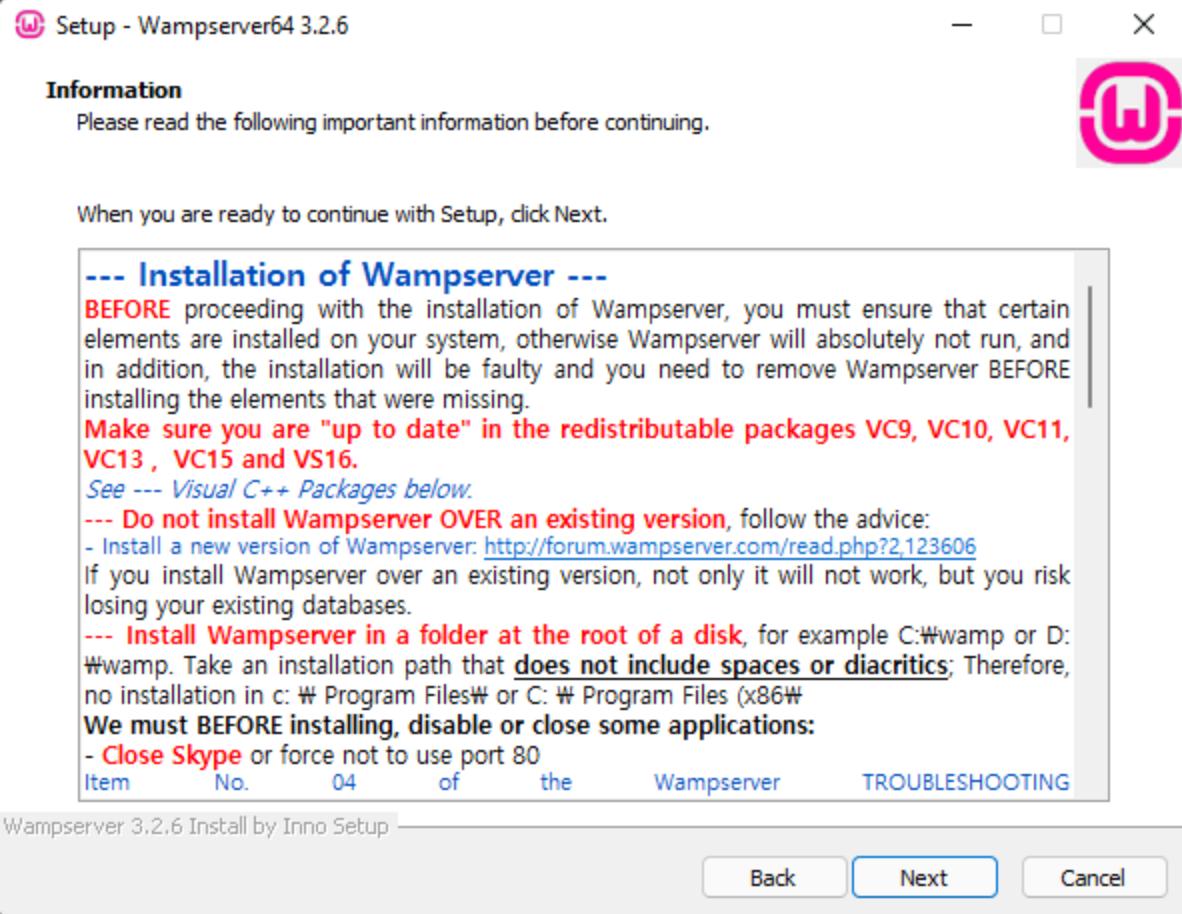
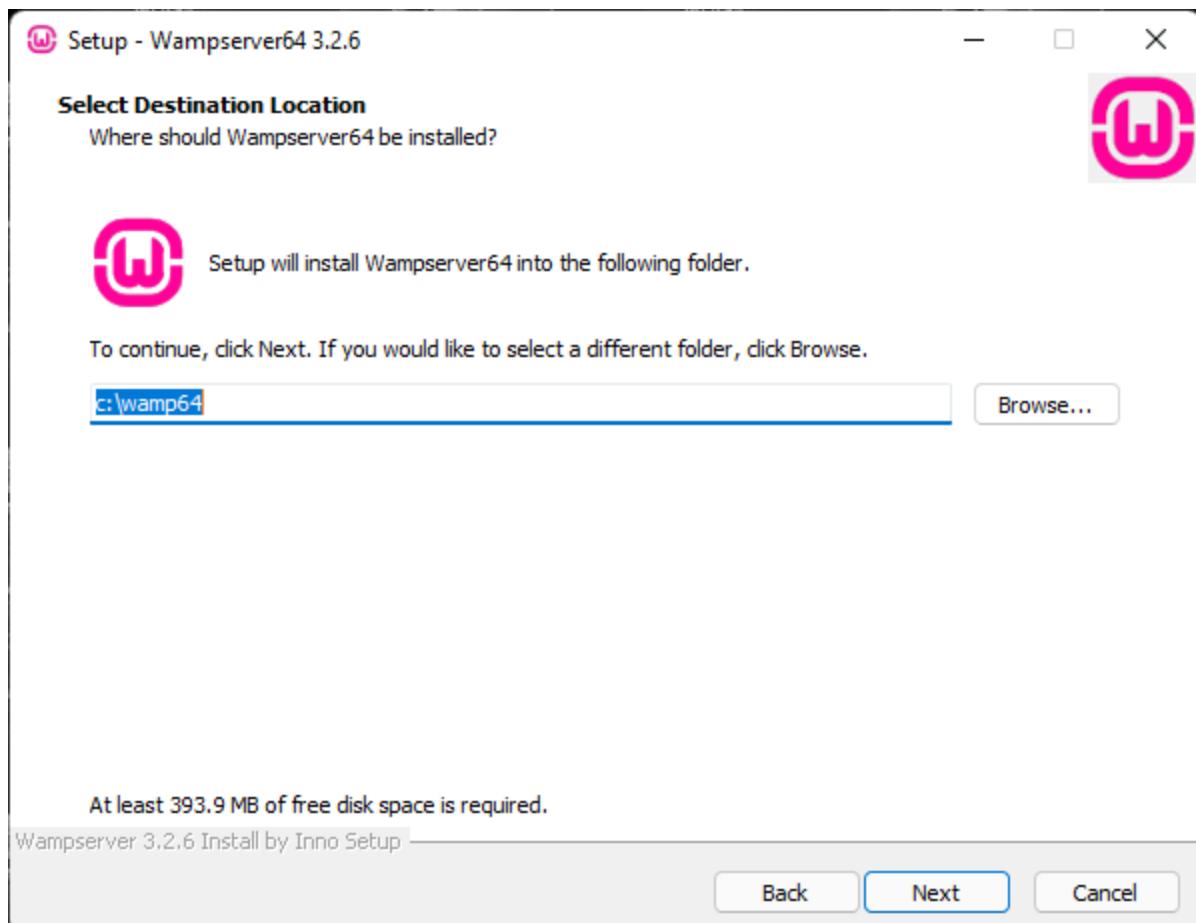


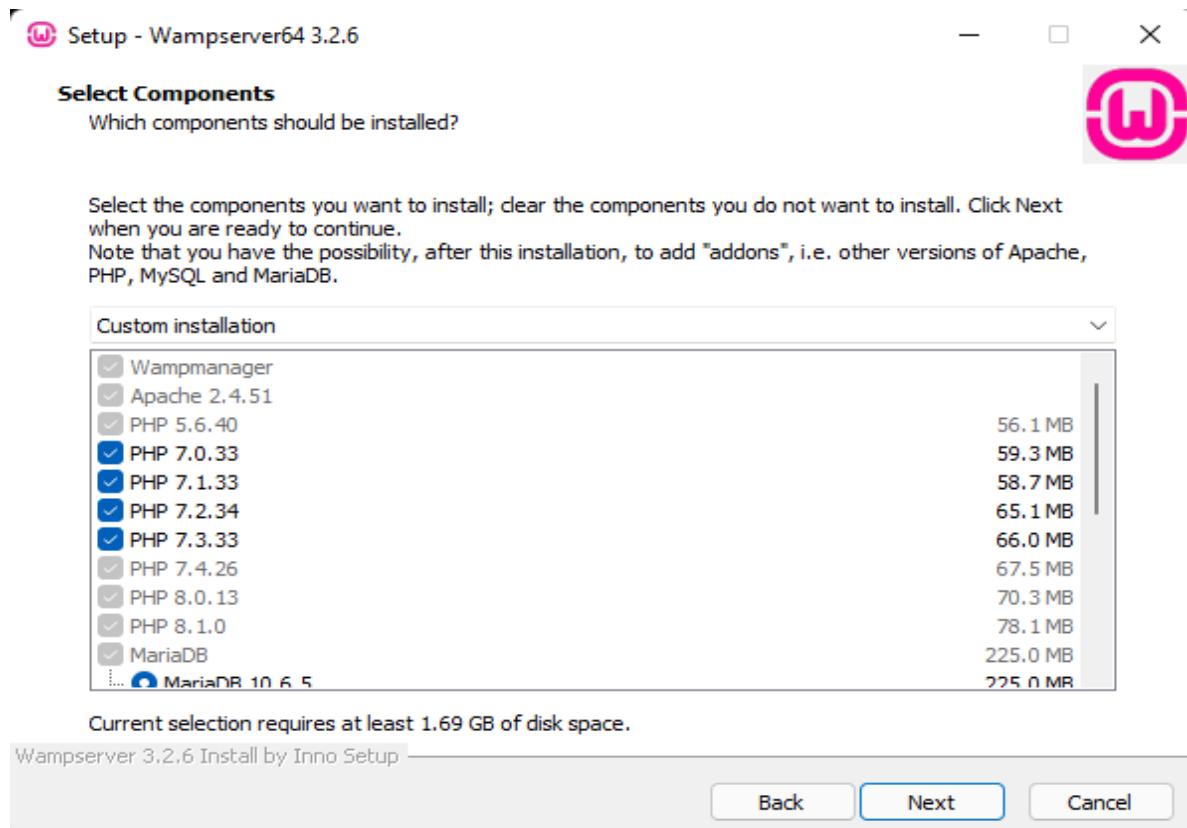
Figure 156: Information on installation

Read and understand on the information given for the installation of WampServer.



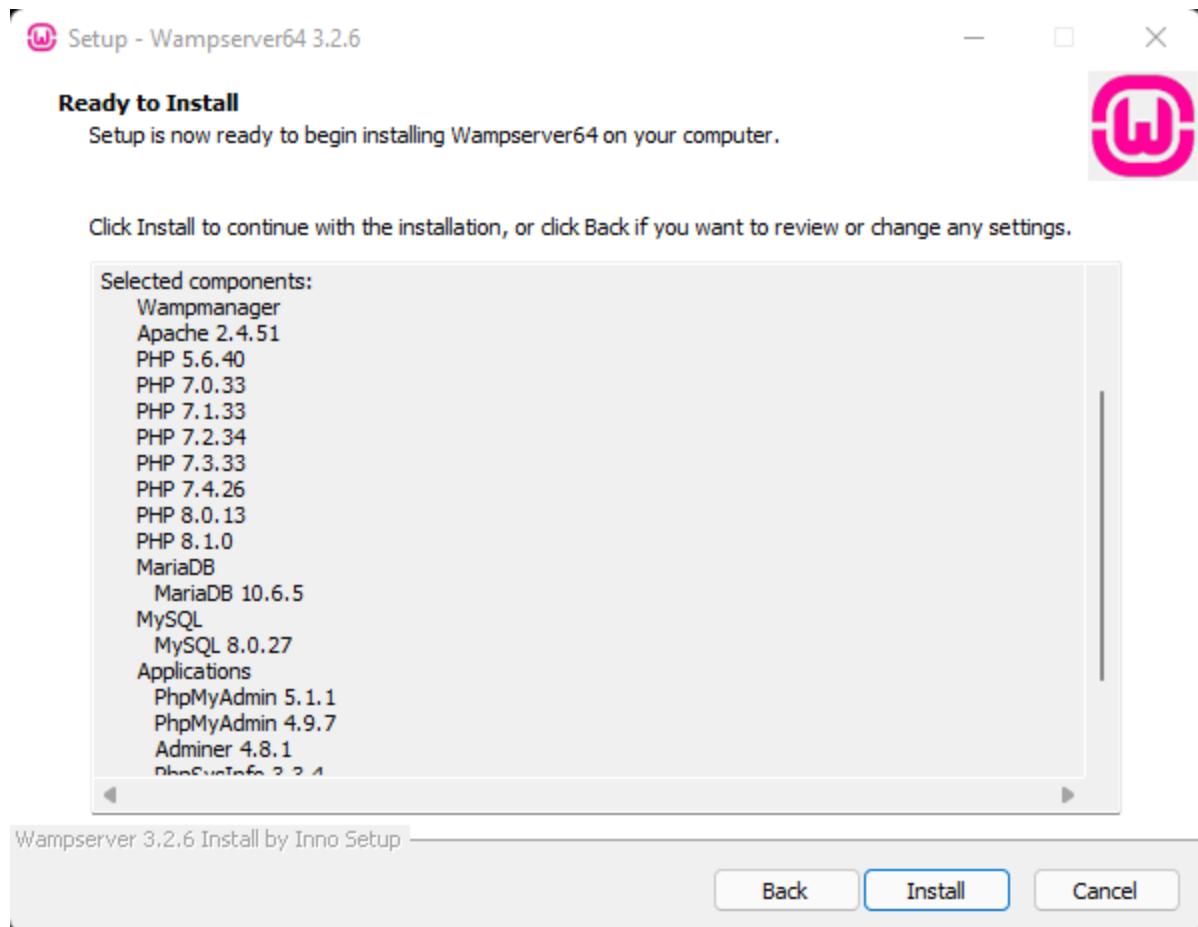
*Figure 157: Select Destination Location*

Select where you would like to install the WampServer folder and remember where it will be placed.



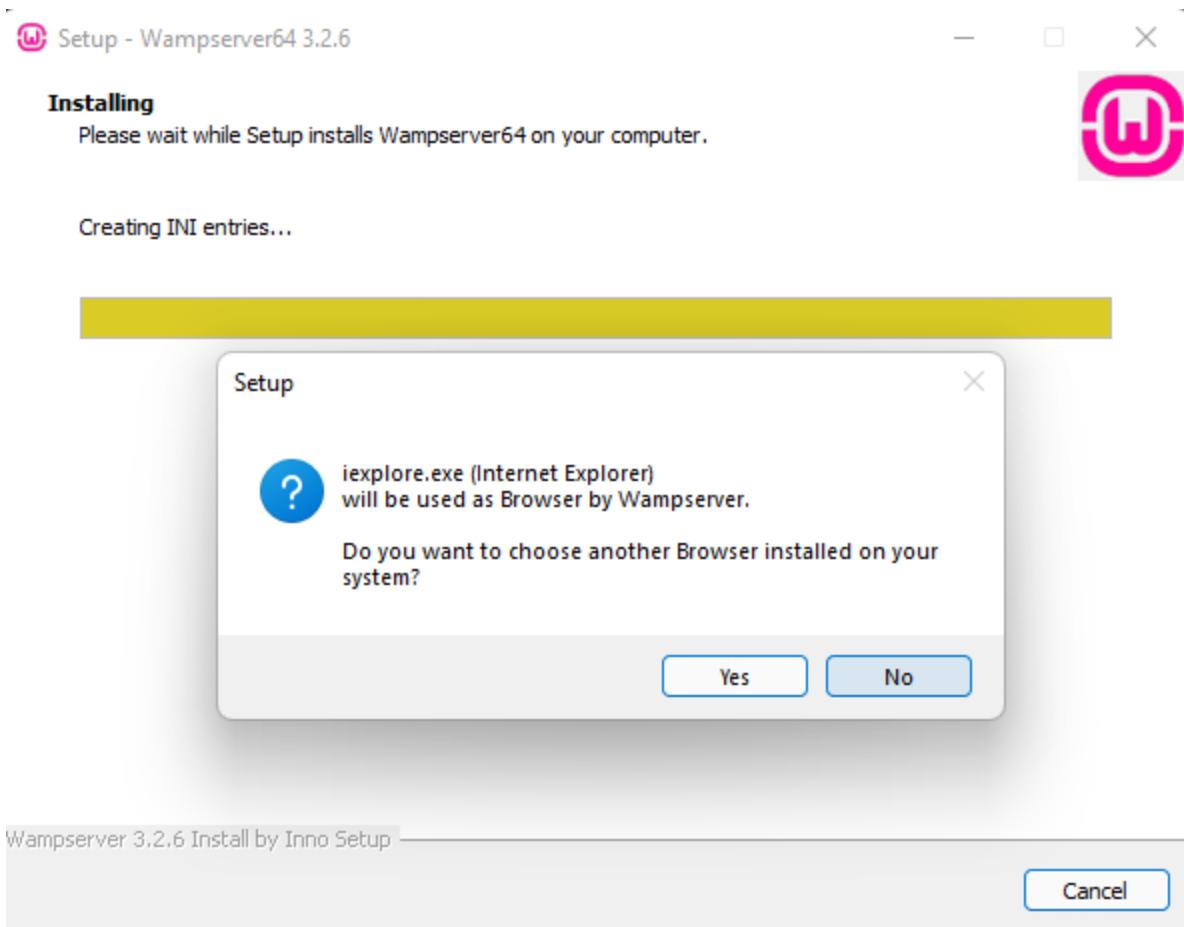
*Figure 158: Select Components*

Select all components that you would like to install onto the WampServer and click on next.



*Figure 159: Ready to Install Component*

These are all the components that will be installed into your WampServer. If you would like to make changes then click on back or just proceed by clicking on install.



*Figure 160: Browser Setup*

Once the installation is completed, a popup message asks for a browser to open your WampServer. If you would like to maintain the browser given then click on no and would like to change then click on yes.

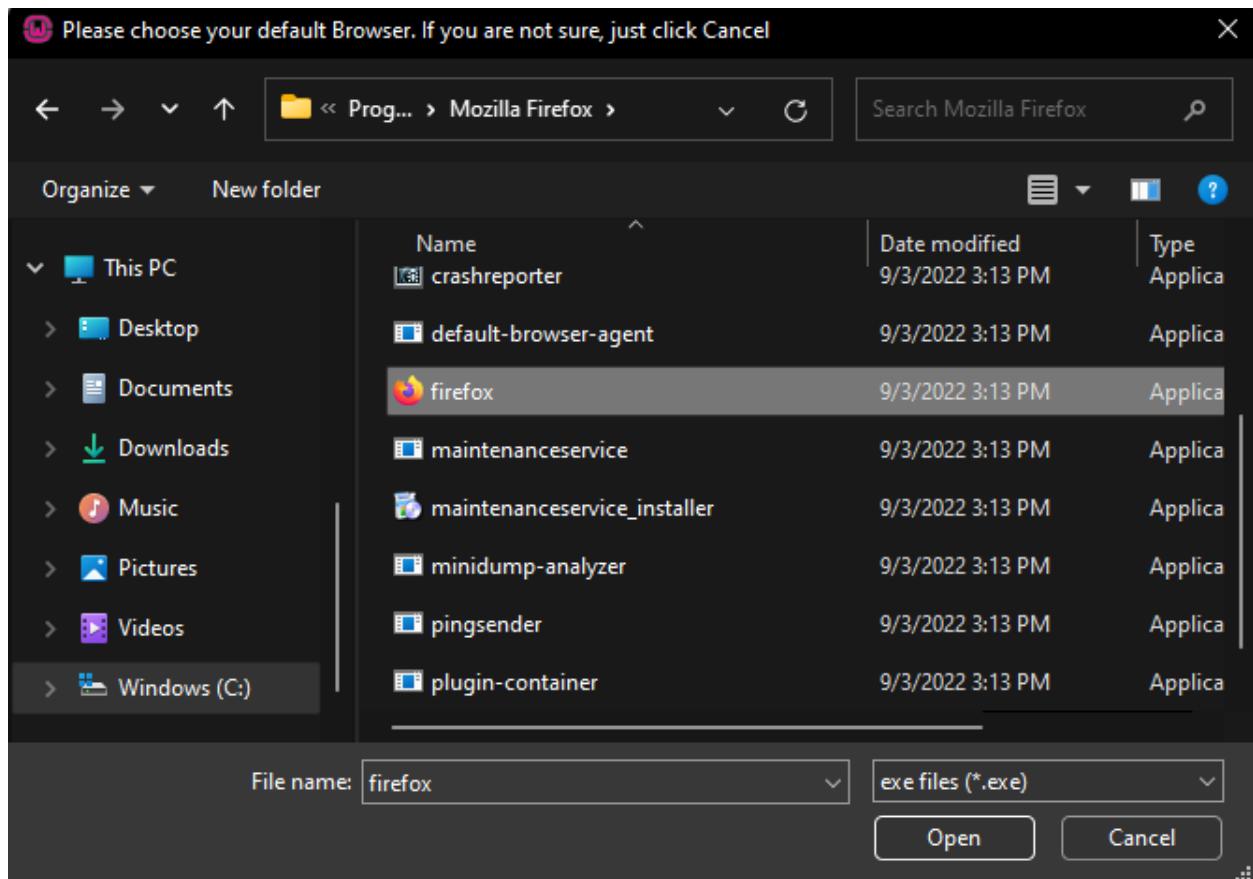
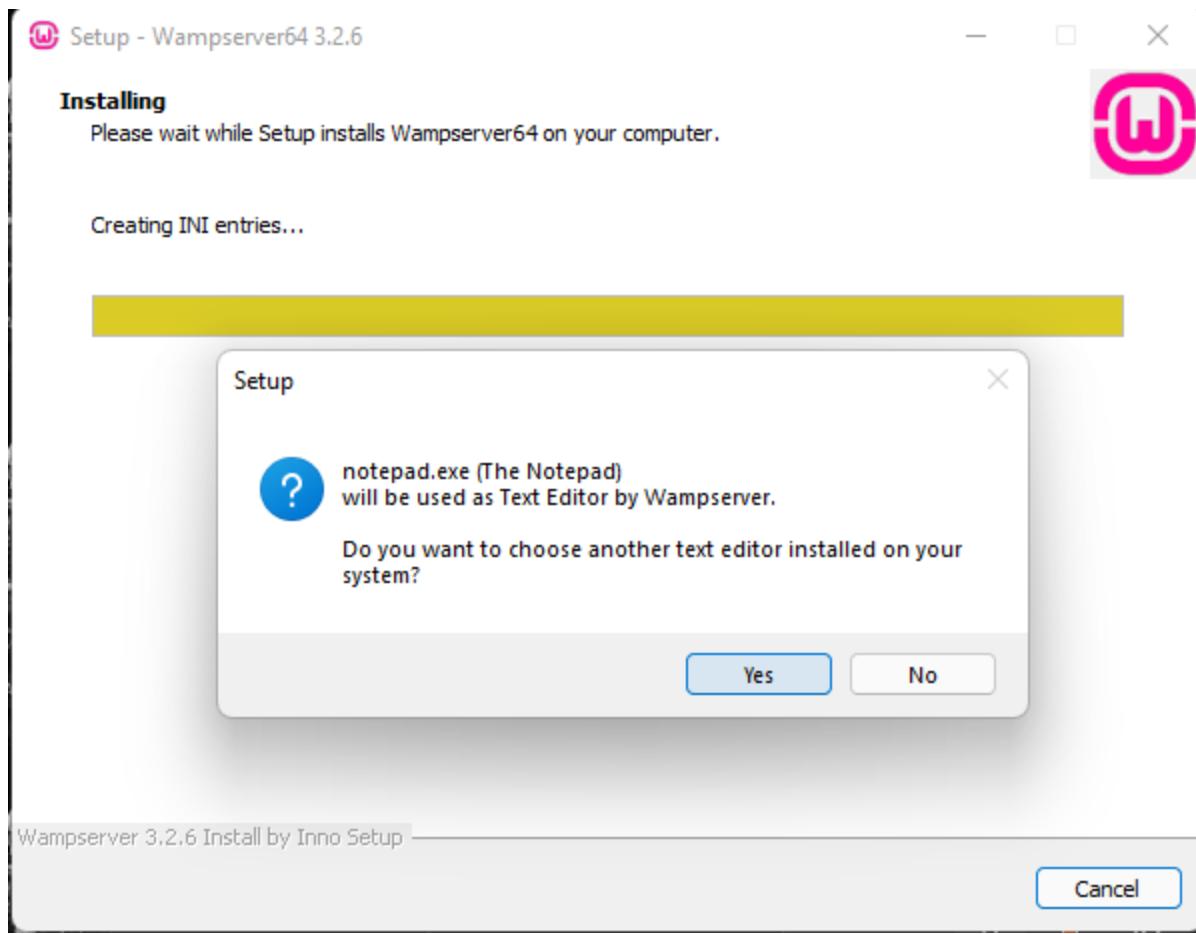


Figure 161: Browser Selection

Select your option of browser for your WampServer if you clicked on yes to change your browser.



*Figure 162: Text Editor Setup*

The next popup message is on the text editor where you can choose the given text editor. If you would like to change then click yes or else click on no and continue the installation.

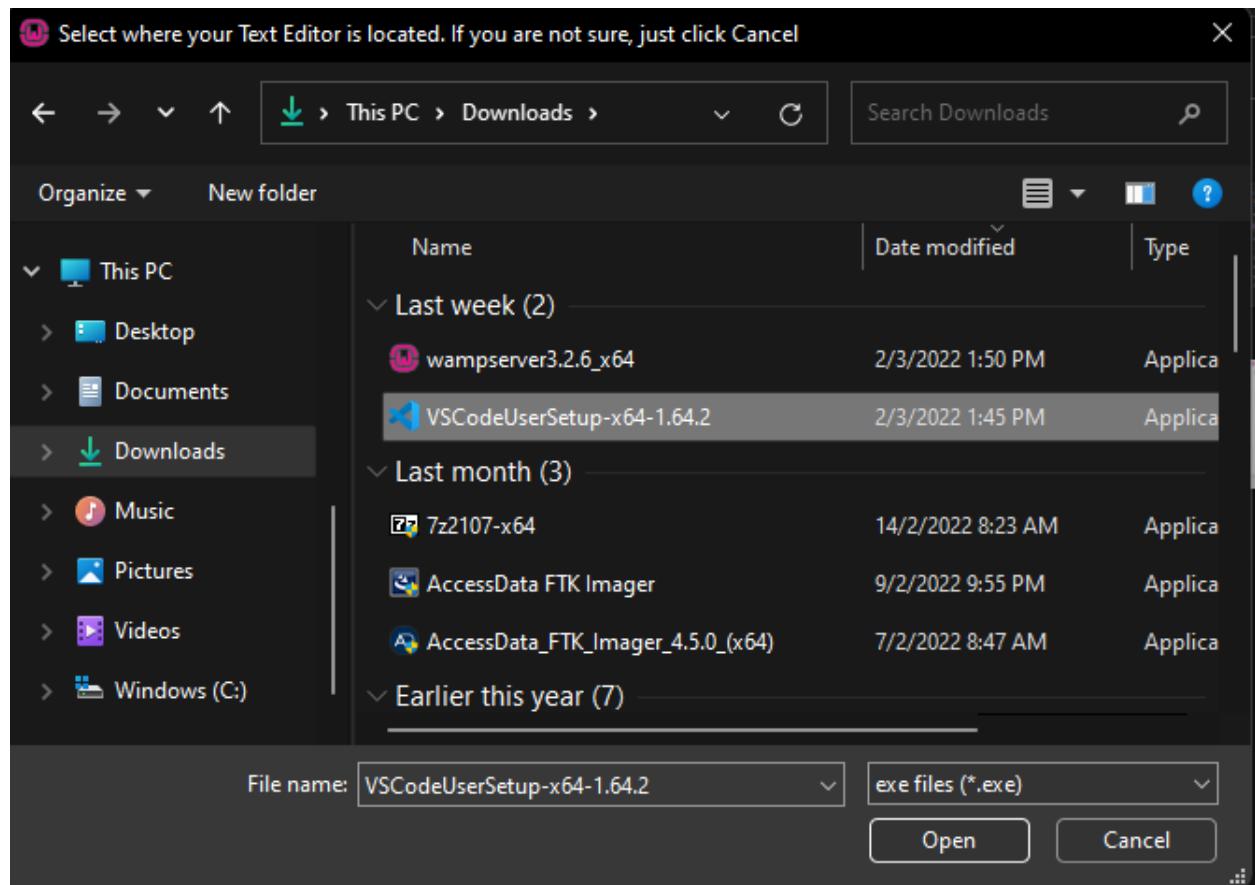


Figure 163: Text Editor Selection

Select your option of text editor for your WampServer if you clicked on yes to change your text editor.

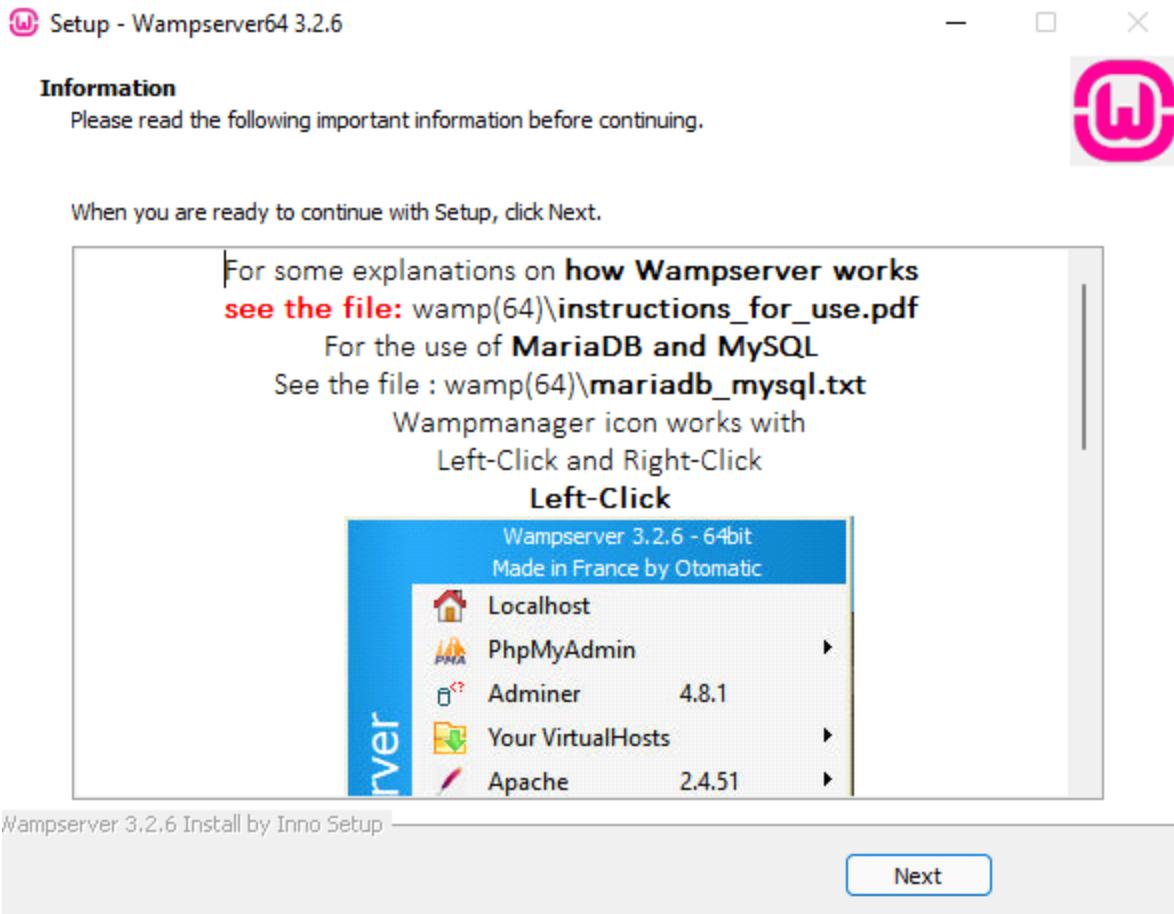
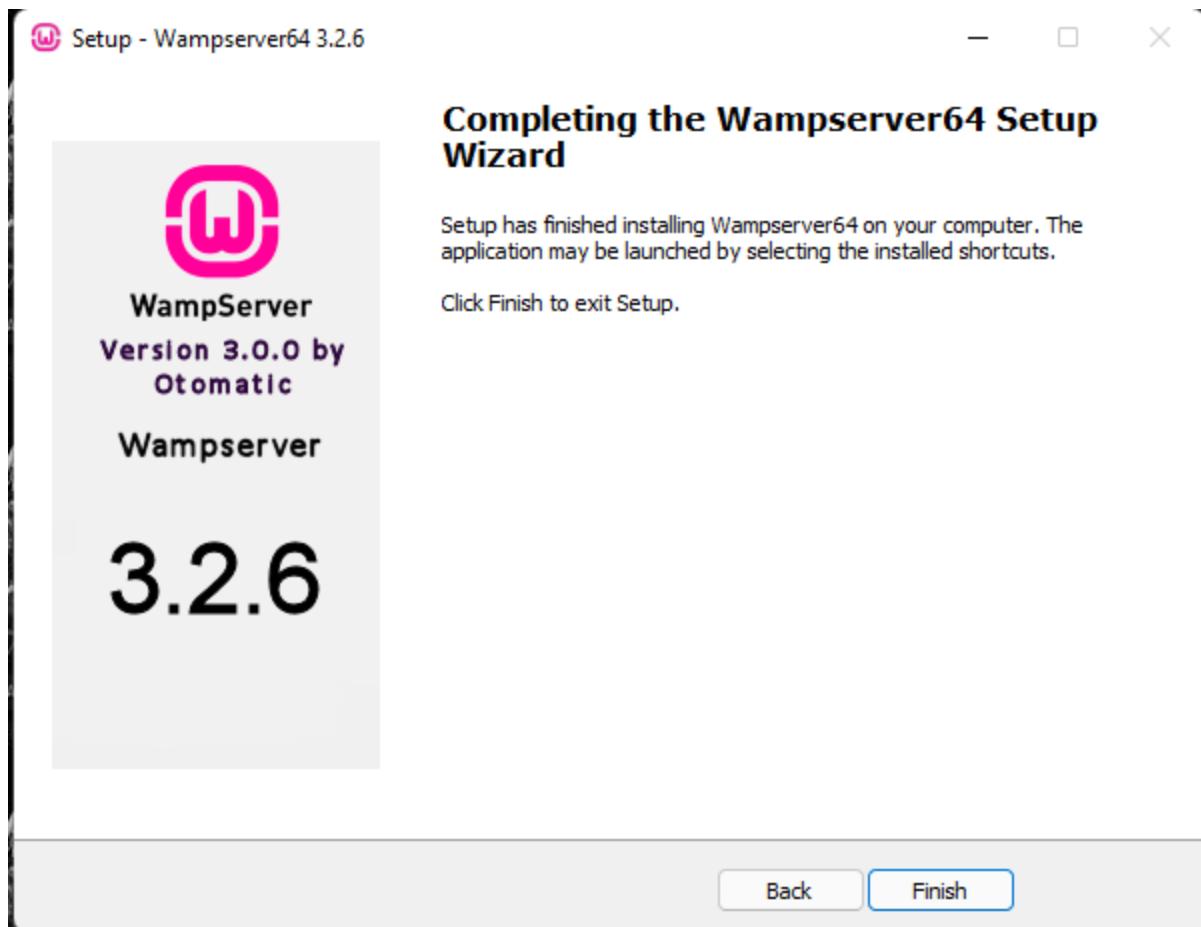


Figure 164: Information of WampServer

The above figure shows the functions and tools available on the left-click and right-click. To continue on setting the WampServer click on next.



*Figure 165: Complete Installation*

Click on finish to complete the installation of WampServer onto your desktop or laptop. Now you can start using WampServer.

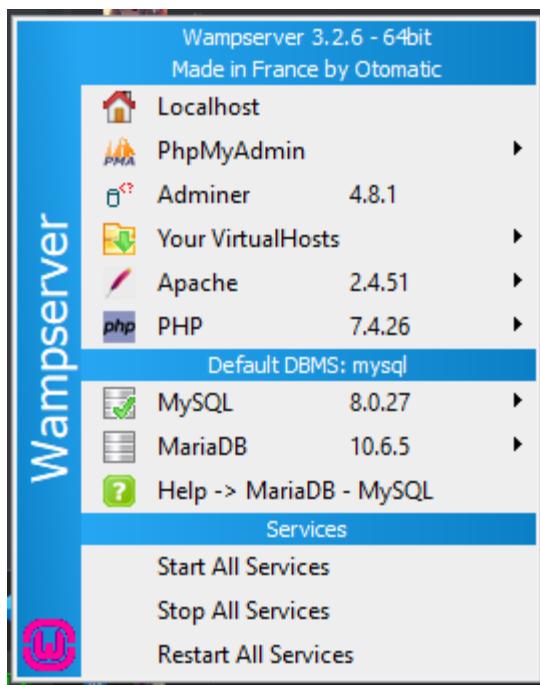


Figure 166: Left-Click on WampServer

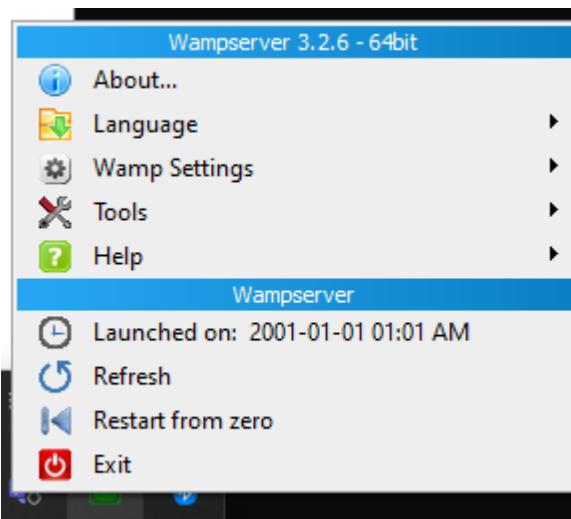


Figure 167: Right-Click on WampServer

The above two figures show the panel of WampServer when the user left-click and right-click to view the functions and tools of the server.

## 21.2 Guide to Find www folder, localhost on browser, port number

### 21.2.1 www Folder

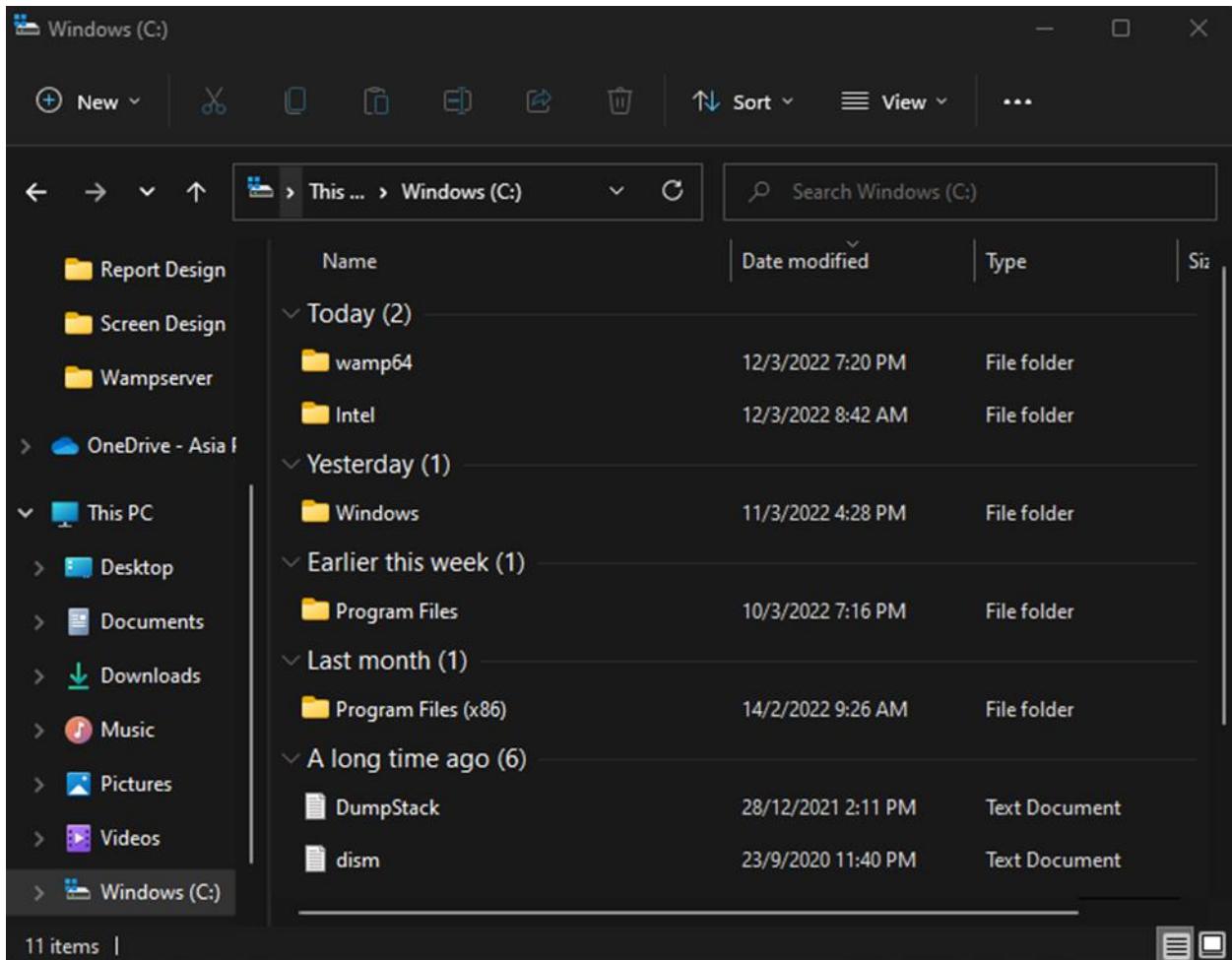


Figure 168: Folder of wamp64 after installation completed.

Search for wamp64 folder after the successfully installed the WampServer. In this case the wamp64 folder is in the Windows drive. After locating the folder click on the said folder to view the content.

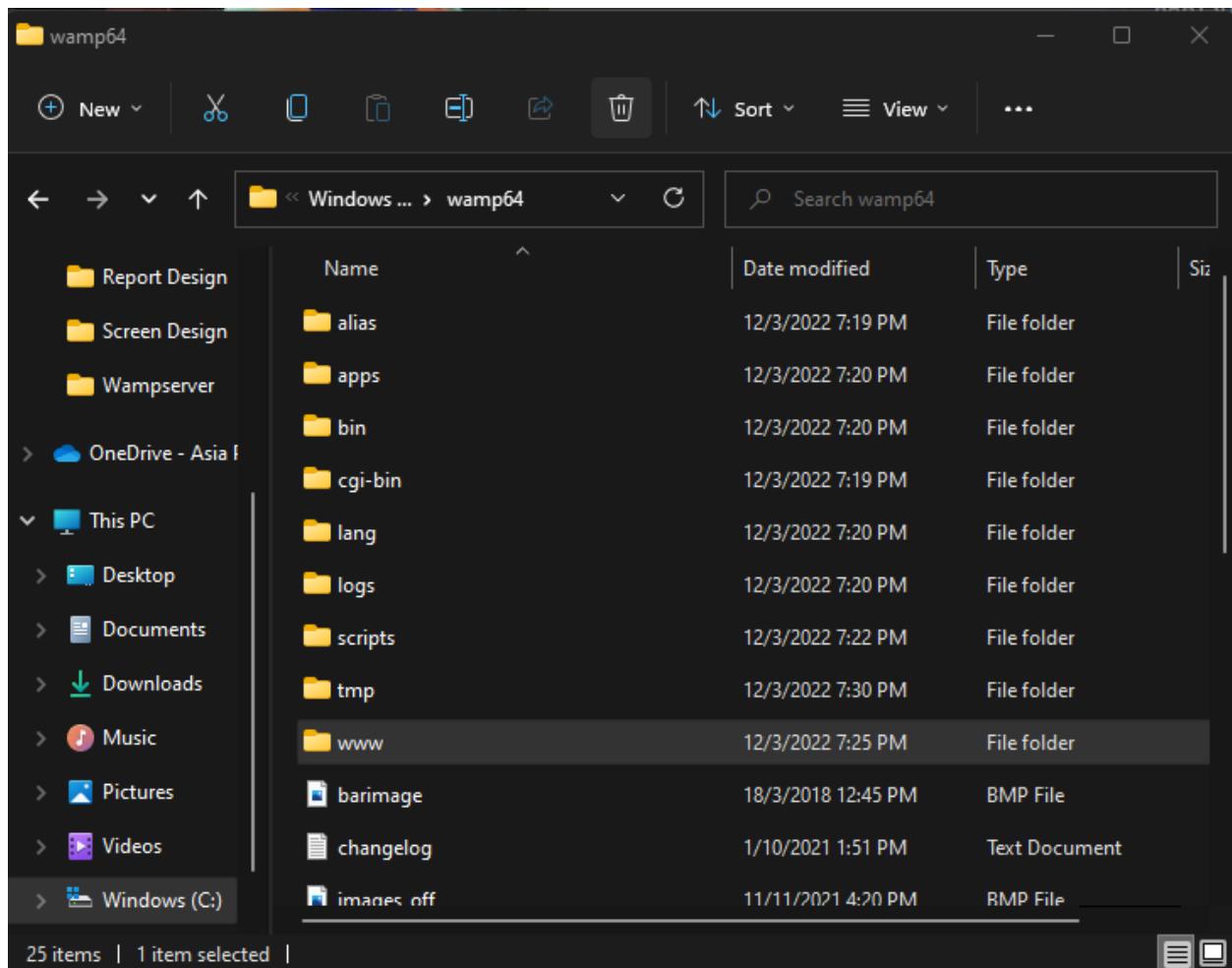


Figure 169: Locate www in folder

Find for the www folder in the wamp64 folder to store the codes for the creation of Valley Clinic Management System.

### 21.2.2 Localhost on browser

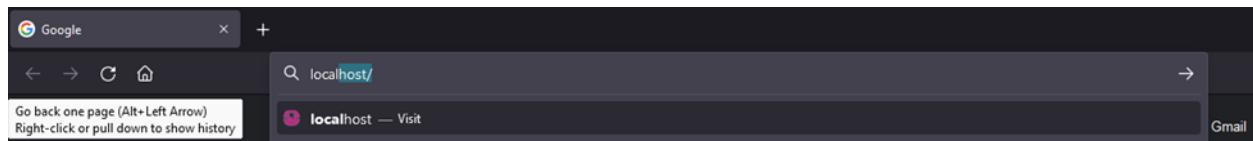


Figure 170: Find localhost

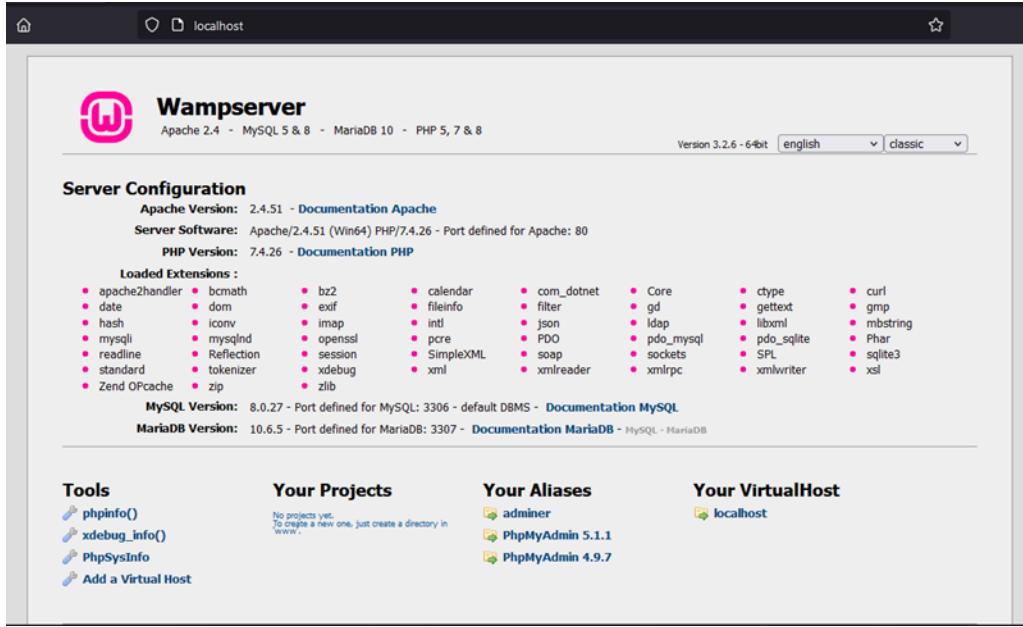


Figure 171: Main page of Localhost

Open your selection of browser and type in localhost and click enter. After clicking enter it will bring to the main page of the local host.

### 21.2.3 Port Number

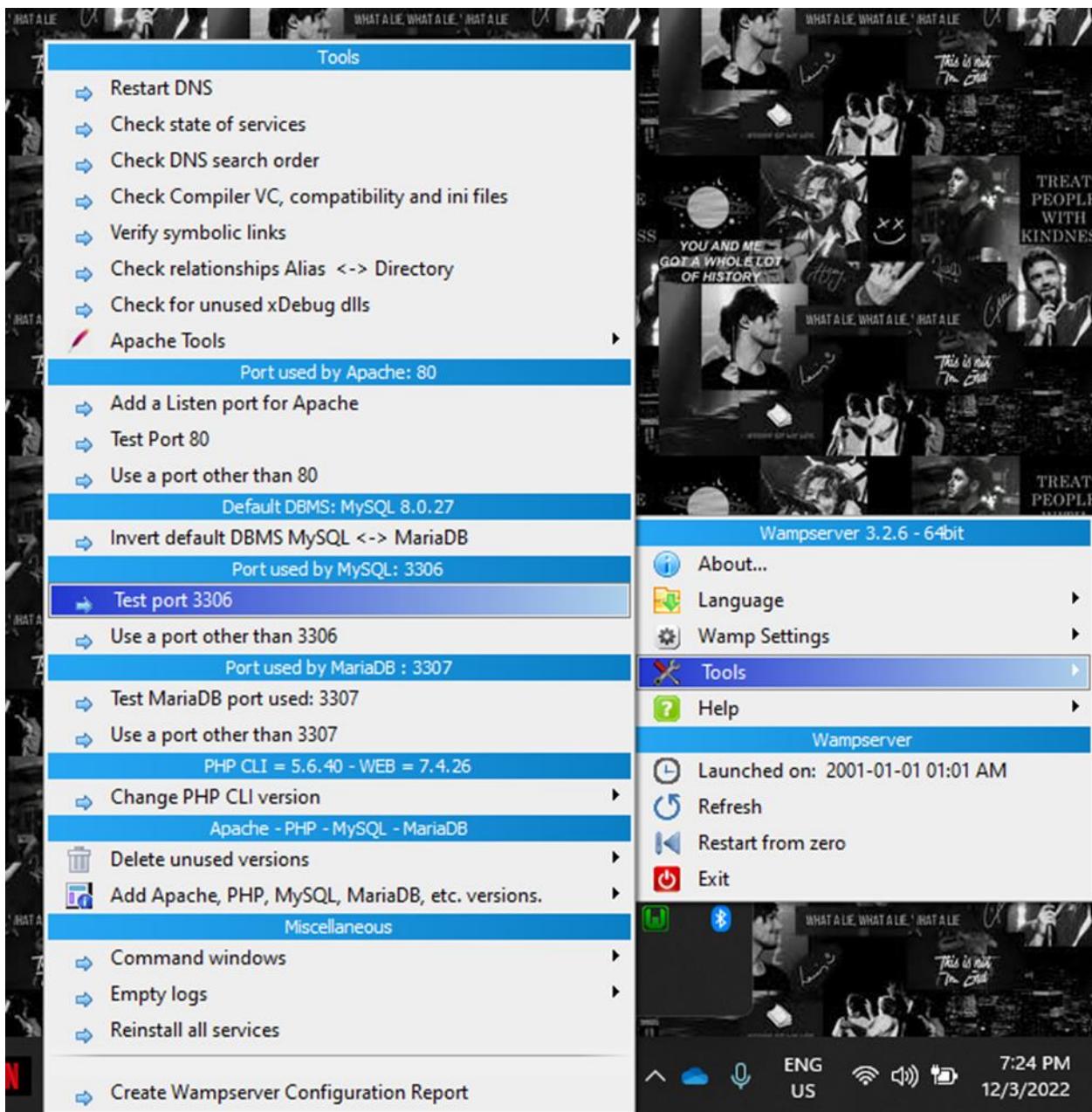
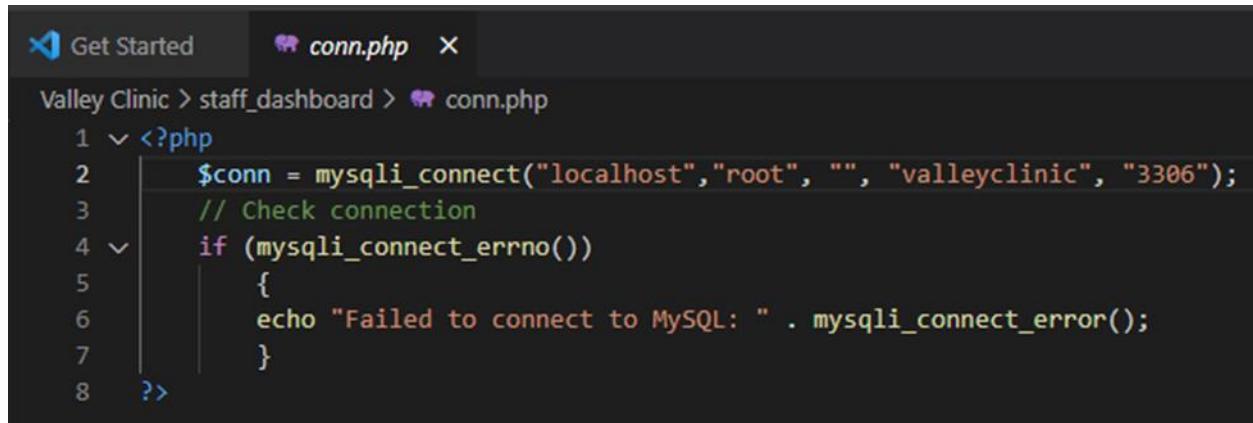


Figure 172: Port Number

To find the port number for the MySQL, right-click on the WampServer logo and click on tools. After clicking on tools another column of information is shown then search for MySQL and see the port number that is used by MySQL which in this case is 3306.

#### 21.2.4 Include Port Number in Code



```
Get Started conn.php X
Valley Clinic > staff_dashboard > conn.php
1 <?php
2     $conn = mysqli_connect("localhost", "root", "", "valleyclinic", "3306");
3     // Check connection
4     if (mysqli_connect_errno())
5     {
6         echo "Failed to connect to MySQL: " . mysqli_connect_error();
7     }
8 ?>
```

Figure 173: Port Number in Code

After knowing the port number, create a conn.php file and connect the MySQL to the source code. The above figure shows on how to connect the MySQL to the codes of this system.

## 21.3 Guide to PhpMyAdmin

### 21.3.1 Login to PhpMyAdmin

phpMyAdmin  
Welcome to phpMyAdmin

Language: English

Log in

Username: root

Password:

Server choice: MySQL

Go

Figure 174: Login Page of PhpMyAdmin

After opening the localhost, click on phpMyAdmin under Your Aliases which will then bring to the log in page. To log in click on go.

Figure 175: Main Page of PhpMyAdmin

After logging in, the main page of phpMyAdmin will be shown as above.

### 21.3.2 Create Database

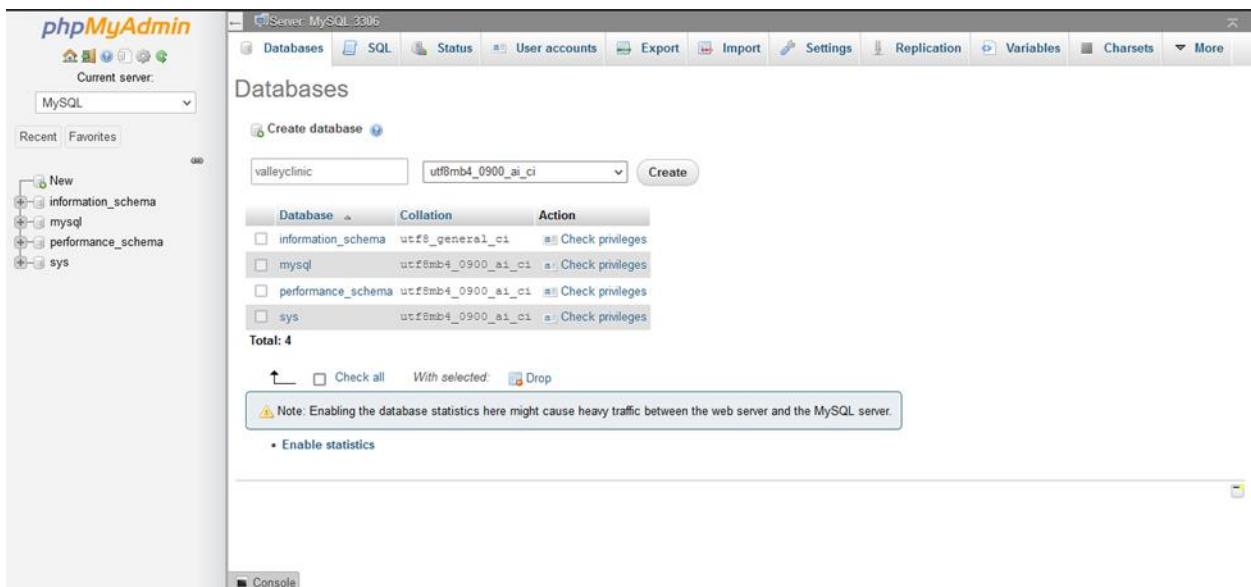


Figure 176: Create Database

Click on the databases tab and create a database by giving it the desired name. After filling out the name click on the create button and the creation of the database is completed.

### 21.3.3 Import Database

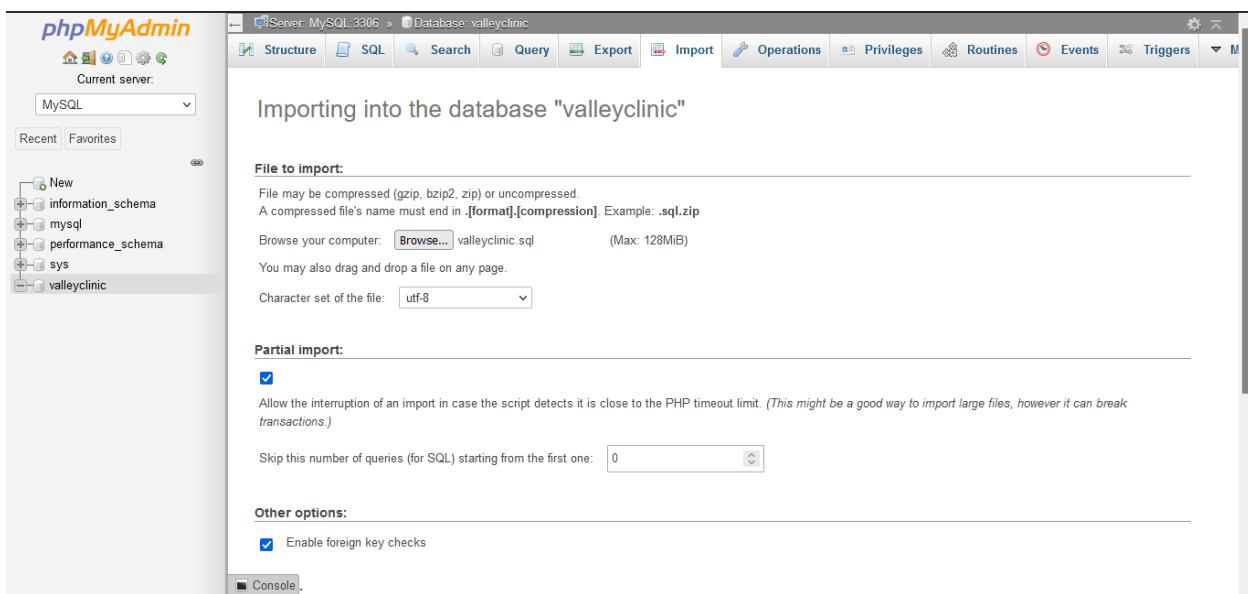


Figure 177: Import Database

To import any data to the database, click on the database created from the list of database on the left navigation panel. This will then bring out the main page of the database. Next, click on the import tab to import any data to the database. After selecting the required file, scroll to the bottom and click on the go button. Hence, all the data from the file will be uploaded into the database.

## 21.4 Minute of Meetings

**Table 1: 1<sup>st</sup> Meeting**

<b>Date</b>	14/1/2022
<b>Time</b>	12.00 pm – 1.30 pm
<b>Attendees</b>	HOOI KAI JUN TEO KAI YII SAM ZHENG YEW TAN KAR WEI NISHARINI GANESAN
<b>Agenda</b>	<ol style="list-style-type: none"><li>1. To choose roles for each member</li><li>2. To discuss the topic of our system</li><li>3. To distribute tasks for proposal</li><li>4. To discuss the project background, objectives, solution and scope.</li><li>5. Create proposal document</li></ol>

**Table 2: 2<sup>nd</sup> Meeting**

<b>Date</b>	21/1/2022
<b>Time</b>	12pm - 1pm
<b>Attendees</b>	HOOI KAI JUN TEO KAI YII SAM ZHENG YEW TAN KAR WEI NISHARINI GANESAN
<b>Agenda</b>	<ol style="list-style-type: none"><li>1. Check previous work</li><li>2. Do corrections for System Hierarchy Chart</li><li>3. Discuss Context Diagram, DFD &amp; ERD</li></ol>

**Table 3: 3<sup>rd</sup> Meeting**

<b>Date</b>	28/1/2022
<b>Time</b>	12pm - 1pm
<b>Attendees</b>	HOOI KAI JUN TEO KAI YII SAM ZHENG YEW TAN KAR WEI NISHARINI GANESAN
<b>Agenda</b>	1. Follow up on previous work 2. Finalize system hierarchy chart and context diagram 3. DFD Level 0

**Table 4: 4<sup>th</sup> Meeting**

<b>Date</b>	10/2/2022
<b>Time</b>	12pm - 1 pm
<b>Attendees</b>	HOOI KAI JUN TEO KAI YII SAM ZHENG YEW TAN KAR WEI
<b>Agenda</b>	1. Follow up on previous work 2. Finalize the DFD Level 0 & ERD

**Table 5: 5<sup>th</sup> Meeting**

<b>Date</b>	16/2/2022
<b>Time</b>	12pm - 1pm

<b>Attendees</b>	HOOI KAI JUN TEO KAI YII SAM ZHENG YEW TAN KAR WEI NISHARINI GANESAN
<b>Agenda</b>	1. Follow up on previous work 2. Check ERD again. 3. Distribute wireframing tasks

**Table 6: 6<sup>th</sup> Meeting**

<b>Date</b>	25/2/2022
<b>Time</b>	12pm - 1pm
<b>Attendees</b>	HOOI KAI JUN TEO KAI YII SAM ZHENG YEW TAN KAR WEI NISHARINI GANESAN
<b>Agenda</b>	1. Follow up on previous work

**Table 7: 7<sup>th</sup> Meeting**

<b>Date</b>	2/3/2022
<b>Time</b>	12pm -1 pm
<b>Attendees</b>	HOOI KAI JUN TEO KAI YII SAM ZHENG YEW TAN KAR WEI NISHARINI GANESAN
<b>Agenda</b>	1. Follow up on previous work 2. Discuss ERD

**Table 8: 8<sup>th</sup> Meeting**

<b>Date</b>	8/3/2022
<b>Time</b>	12pm - 1pm
<b>Attendees</b>	HOOI KAI JUN TEO KAI YII SAM ZHENG YEW TAN KAR WEI NISHARINI GANESAN
<b>Agenda</b>	<ol style="list-style-type: none"><li>1. Follow up progress</li><li>2. Discuss significant source code (CRUD) and the part that need references</li><li>3. Discuss test case</li><li>4. Discuss user guide (wireframe + actual)</li><li>5. Discuss acknowledgement</li><li>6. Discuss conclusion</li></ol>

**Table 9: 9<sup>th</sup> Meeting**

<b>Date</b>	16/3/2022
<b>Time</b>	12pm - 1pm
<b>Attendees</b>	HOOI KAI JUN TEO KAI YII SAM ZHENG YEW TAN KAR WEI NISHARINI GANESAN
<b>Agenda</b>	<ol style="list-style-type: none"><li>1. Follow up progress</li><li>2. Gather source code</li></ol>

	3. Gather database 4. Check documentation
--	--

**Table 10: 10<sup>th</sup> Meeting**

<b>Date</b>	23/3/2022
<b>Time</b>	12pm - 1pm
<b>Attendees</b>	HOOI KAI JUN TEO KAI YII SAM ZHENG YEW TAN KAR WEI NISHARINI GANESAN
<b>Agenda</b>	1. Follow up progress 2. Update references 3. Check documentation

## 21.5 Milestones / Project Progress

Milestones / Project Progress Group 5								
Task Name	Start Date	End Date	Priority	Assigned To (Full Name)	Completion Status	Lecturer Comments	Group Leader Comments	Document Link
1.0 Proposal	14/3/2022	7/2/2022	High		Completed			
1.1 Cover Page	14/3/2022	25/1/2022	High	NISHARINI GANESAN	Done			<a href="#">SOFTWARE DEVELOPMENT PROJECT PROPOSAL.docx</a>
1.2 Table of Content	14/3/2022	25/1/2022	High	NISHARINI GANESAN	Done			
1.3 Workload Matrix	14/3/2022	25/1/2022	High	TAN KAR WEI	Done			
1.4 Introduction	14/3/2022	25/1/2022	High		Completed			
1.4.1 Project Background	14/3/2022	25/1/2022	High	HOOI KAI JUN	Done			
1.4.2 Problem Context	14/3/2022	25/1/2022	High	TEO KAI YII	Done			
1.4.3 Proposed Solution	14/3/2022	25/1/2022	High	TAN KAR WEI	Done			
1.4.4 Project Objectives	14/3/2022	25/1/2022	High	NISHARINI GANESAN	Done			
1.4.5 Project Scope	14/3/2022	25/1/2022	High	SAM ZHENG YEW	Done			
1.5 Project Plan	14/3/2022	25/1/2022	High		Completed			
1.5.1 System Development Methodology	14/3/2022	25/1/2022	High	HOOI KAI JUN	Done			
1.5.2 Project Gantt Chart	14/3/2022	25/1/2022	High	TEO KAI YII	Done			
1.6 System Hierarchy Chart	14/3/2022	25/1/2022	High	SAM ZHENG YEW	Done			
1.7 Conclusion	14/3/2022	25/1/2022	High	NISHARINI GANESAN	Done			
2.0 Main Document					Completed			<a href="#">SDP Assignment(19-3-2022).docx</a>
2.1 Acknowledgement	16/3/2022	23/3/2022	High	TEO KAI YII	Done			
2.2 Abstract	16/3/2022	23/3/2022	High	NISHARINI GANESAN	Done			
2.3 Workload Matrix	16/3/2022	23/3/2022	High	TEO KAI YII	Done			
2.1 Context Diagram	25/1/2022	9/2/2022	High	HOOI KAI JUN & TEO KAI YII	Done			
2.2 Data Flow Diagrams Level 0	25/1/2022	9/2/2022	High	TAN KAR WEI & NISHA	Done			
2.3 Data Flow Diagrams Level 1	25/1/2022	9/2/2022	High	(Individual)	Done			
2.4 Data Dictionary	25/1/2022	16/2/2022	High	(Individual)	Done			
2.5 Entity Relationship Diagram	25/1/2022	16/2/2022	High	SAM ZHENG YEW	Done			
2.6 Pseudocode or Flowcharts	25/1/2022	16/2/2022	High	(Individual)	Done			
2.7 Screens Design and User Manual	16/2/2022	2/3/2022	High	(All)	Done			
2.8 Report Design and User Manual	2/3/2022	16/3/2022	High	(All)	Done			
2.9 Test Plan (Unit Testing)	8/3/2022	16/3/2022	High	(All)	Done			
2.10 Significant Source Code	8/3/2022	16/3/2022	High	HOOI KAI JUN, TEO KAI YII, TAN KAR WEI, SAM ZHENG YEW	Done			
2.11 Conclusion	8/3/2022	23/3/2022	High	(All)	Done			
2.12 References	16/3/2022	23/3/2022	High	(All)	Done			
2.13 Appendix	16/3/2022	23/3/2022	Low	TEO KAI YII, NISHARINI GANESAN	Done			

Figure 191: Milestones/Project Progress