WINTAN HOSPITAL

RIHAM AHAMED ABDUL RAHEEM

HND COMPUTING IDM

Contents

RESEARCH ON EXISTING ONLINE PATIENT APPOINTMENT SYSTEM 2

WINTAN HOSPITAL: PROJECT PLAN (PROJECT SCHEDULE & GANTT CHART) 3

USER & SYSTEM REQUIREMENTS 3

MEDIUM FIDELITY WIREFRAMING FOR US.. 4

Homepage 4

Register 5

Login.. 6

Make Appointments.. 6

WHY SUCH A SIMPLE DESIGN? 7

TECH STACK 8

USER DOCUMENTATION 9

Homepage 9

Compare, contrast and evaluation of design to development 9

Register 10

Compare, contrast and evaluation of design to development 10

Login.. 10

Compare, contrast and evaluation of design to development 11

Profile.. 11

Compare, contrast and evaluation of design to development 11

Appoint a doctor 12

Compare, contrast and evaluation of design to development 12

Your appointments.. 12

Compare, contrast and evaluation of design to development 13

Doctor schedule 13

Compare, contrast and evaluation of design to development 13

EVIDENCE FOR LO3 and LO4.. 14

The project-tree overview 14

Example of a view file (Using PHP, HTML and Javascript).. 15

Other Required Evidence.. 16

TECHNICAL DIFFICULTIES FACED DURING THE DEVELOPMENT 17

TEST SUITE 18

Performance Test 19

QUALITY ASSURANCE 20
OVERALL SUCCESS REVIEW AND POSSIBLE IMPROVEMENTS 21

List of figures

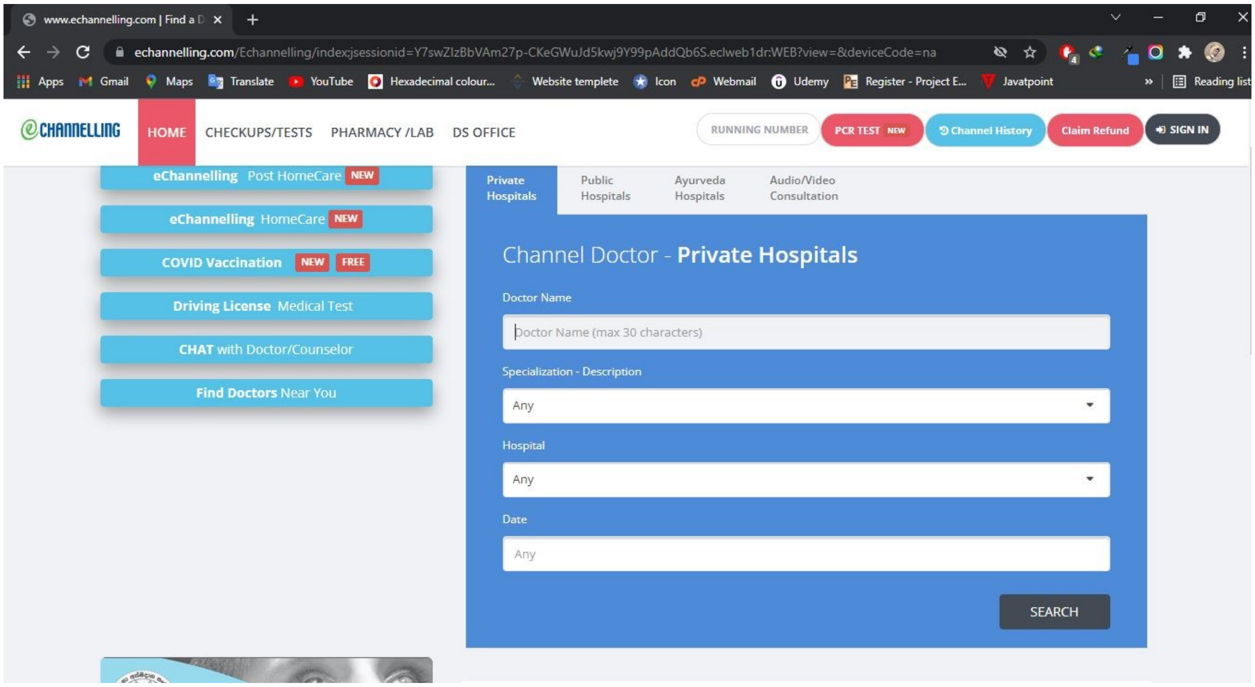
Figure 1: e-channeling website 2
Figure 2: Gantt chart 3
Figure 3: Homepage wireframe 4
Figure 4: Patient registration wireframe 5
Figure 5: Admin login wireframe 6
Figure 6: Make appointments wireframe 6
Figure 7: homepage wintan hospital 9
Figure 8: Registration page wintan hospital 10
Figure 9: Login page Wintan hospital 10
Figure 10: Profile page Wintan hospital 11
Figure 11: Appoint a doctor wintan hospital 12
Figure 12: Appointments page wintan hospital 12
Figure 13: Doctor schedule page wintan hospital 13
Figure 14: project tree 14
Figure 17: Other database data view evidence 16
Figure 18: Date process 18

List of Tables

Table 1: Test report table 19

RESEARCH ON EXISTING ONLINE PATIENT APPOINTMENT SYSTEM

Choice of online patient appointment system : e-channeling (https://www.echannelling.com/Echannelling/index?view=&deviceCode=na)



Channel Doctor - **Private Hospitals**

Doctor Name: Doctor Name (max 30 characters)

Specialization - Description: Any

Hospital: Any

Date: Any

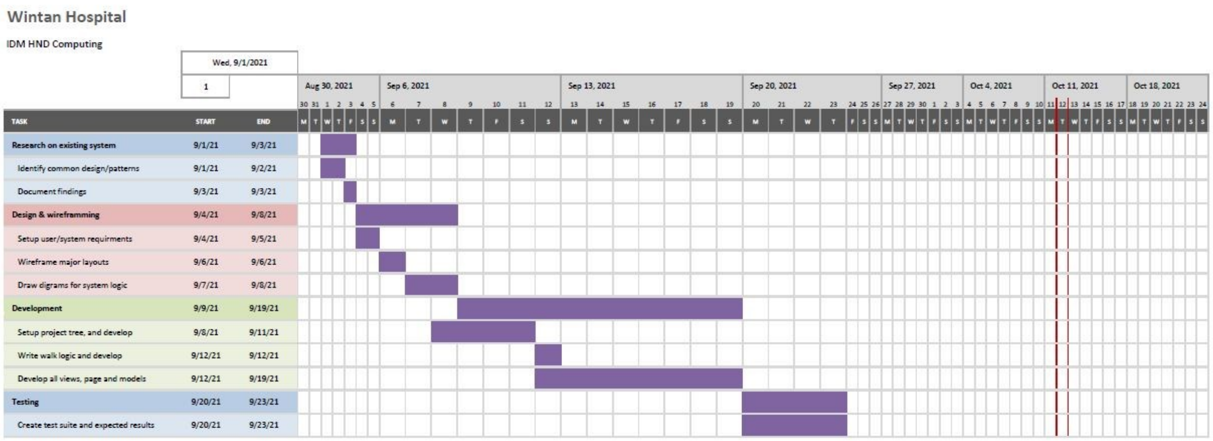
Figure 1: e-channeling website

The reason I chose e-channeling online patient appointment system is because it is extremely similar to the same basic requirements Wintan Hospital requires. e-channeling is known to be one the pioneers in the hospital industry, and as such the conclusions I draw from this research will be quite valid.

Figure 1, shows the patient appointment system e-channeling has. From first glance it becomes quite obvious that there is no registration or user access control setup. The form it presents asks some pretty sensitive information like for example the Payee Address.

In terms of design complexity and functionality, the patient appointment system of e-channeling seems to be quite adept. The colors used aren't difficult on the eyes, there is proper contrast between font and the colors used which is important for a website that attracts patients that may have sight problems. It also seems to be a mobile-first design, as such it is responsive across screens.

WINTAN HOSPITAL: PROJECT PLAN (PROJECT SCHEDULE & GANTT CHART)



Wintan Hospital

IDM HND Computing

Wed, 9/1/2021

1

TASK	START	END
Research on existing system	9/1/21	9/3/21
Identify common design/patterns	9/1/21	9/2/21
Document findings	9/3/21	9/3/21
Design & wireframming	9/4/21	9/8/21
Setup user/system requirments	9/4/21	9/5/21
Wireframe major layouts	9/6/21	9/6/21
Draw digrams for system logic	9/7/21	9/8/21
Development	9/9/21	9/19/21
Setup project tree, and develop	9/8/21	9/11/21
Write walk logic and develop	9/12/21	9/12/21
Develop all views, page and models	9/12/21	9/19/21
Testing	9/20/21	9/23/21
Create test suite and expected results	9/20/21	9/23/21

Figure 2: Gantt chart

USER & SYSTEM REQUIREMENTS

Through the research that was done on e-channeling and the assignment scenario, conclusions and assumptions were made, as such the following are the user and system requirements. Do note, however that these requirements may or may not be fulfilled but the best attempt will be made.

User requirements:

- Patients should be capable of appointing a doctor
- Patients should be able to look at doctor schedules
- Patients should be able to see the appointments they've made
- Should not be complex or difficult to get started with
- Colors used must not be hard on the eyes
- The text colors must contrast the background colors

System requirements:

- At least 6 interlinked webpages
- A database to store patient information
- Frontend that will be used to collect data
- A backend that will process form data sent from the frontend
- Security must be ensured during form data processing
- Mobile-first/responsive frontend UI/UX design

MEDIUM FIDELITY WIREFRAMING FOR US

Wireframing was done to design the layouts before being upgraded to meet the user and system needs mentioned above. All user interfaces for collecting data were wireframed, although in the actual development phase there may be more user interfaces that may or may not collect data.

Some UI wireframes don't include implicit form details because I justify this as a decision to be made during development.

Homepage

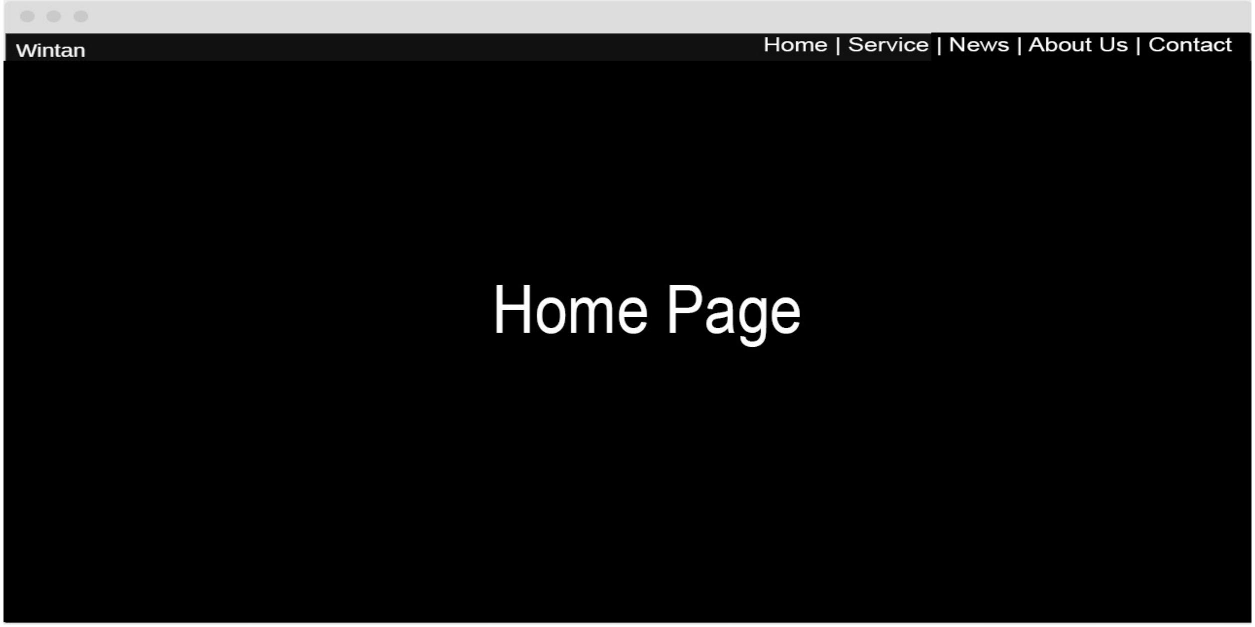


Figure 3: Homepage wireframe

Figure 3, is the wireframe for the homepage. It includes the navigation bar on the top with Home, Service, News, About-us, Contact buttons, a brand name will be set to the right side of the navigation bar. The black box represents an image that will be relevant to this website, and there'll be a bottom Admin login, Doctor login, Patient login and patient registration button are included.

Register

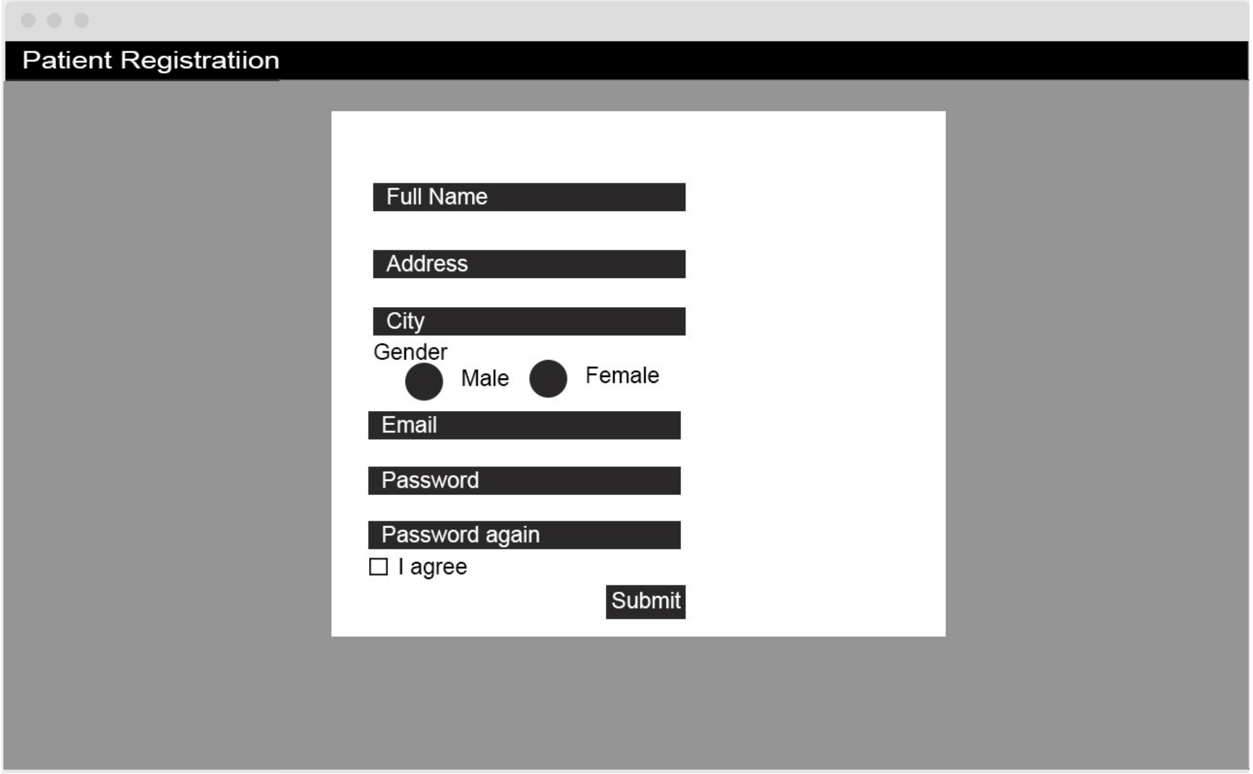


Figure 4: Patient registration wireframe

This will be the register form that contains the first data collection form, that will create an account for the patient. The actual contents of the register form will be decided in the implementation stage. However, there will be 3 distinct components in the register form. The header text, the form and the submit button. and I include sample form also.

Login

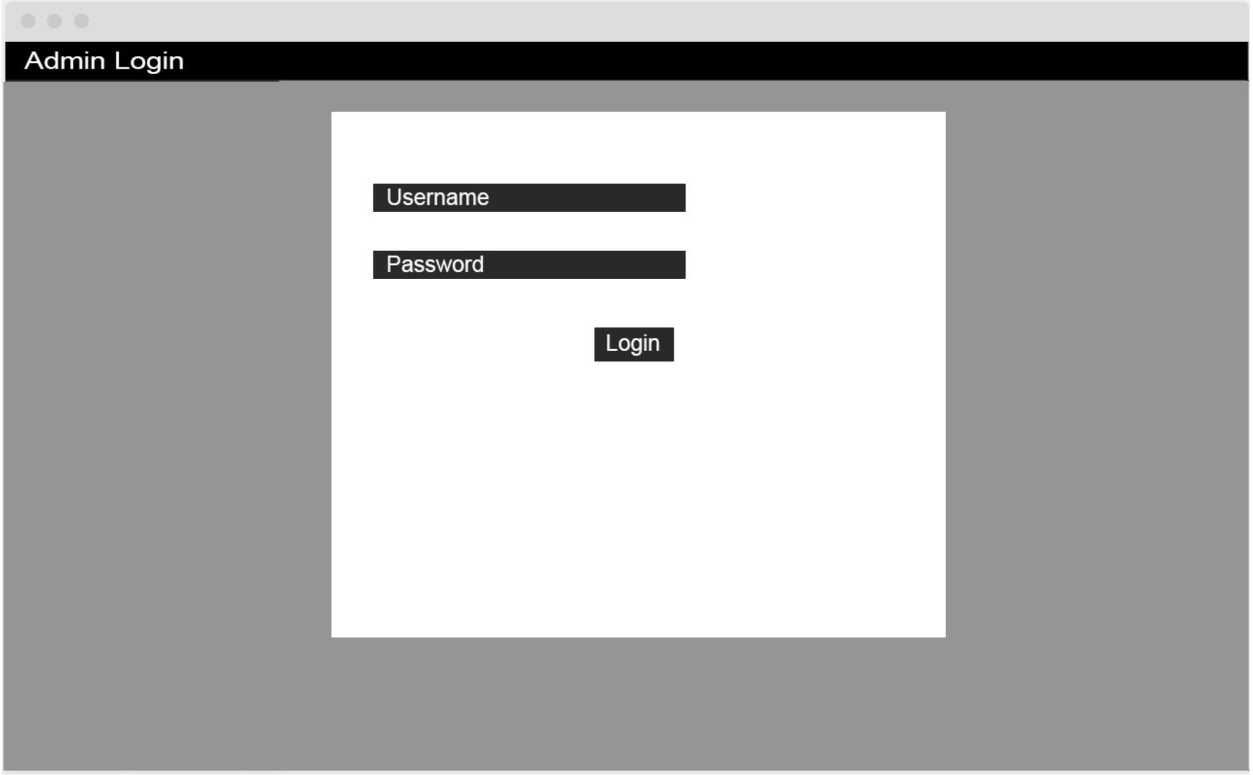


Figure 5: Admin login wireframe

This will be the login form. The 3 main components are the same. However, the function of the login page will be significantly different and I include sample form also.

Make Appointments

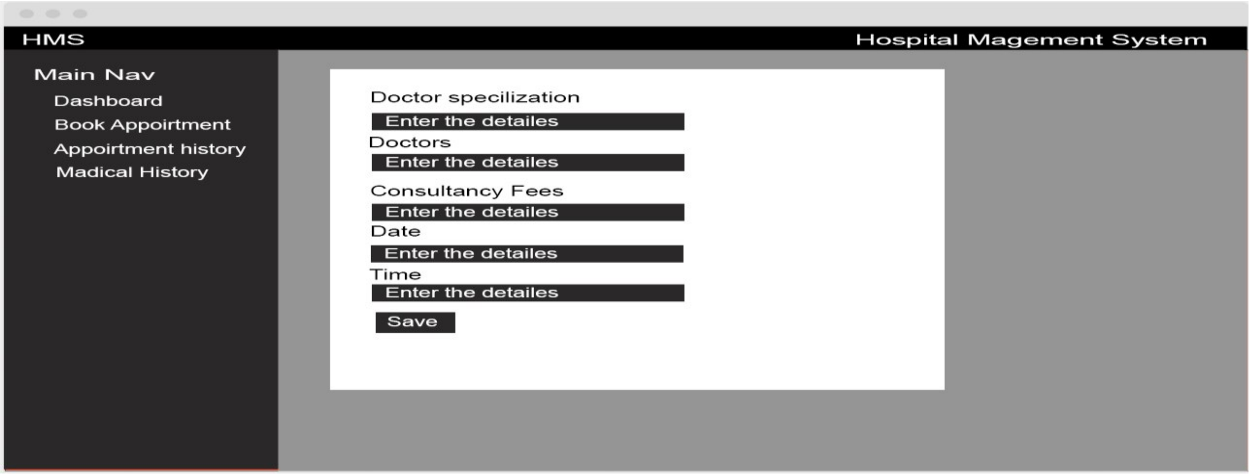


Figure 6: Make appointments wireframe

This will be the core of the patient appointment system. This page will only be visible once a user has registered and then logged in, there will be 3 main components in this form, the header text, left navigation bar the form itself and the submit button.

The navigation bar will have updated showing 4 distinct components Dashboard, Book Appointment, Appointment history, and medical history

A date picker and a drop-box that will have all the doctors available for appointing.

WHY SUCH A SIMPLE DESIGN?

The reason is because most people in Sri Lanka are technologically challenged. The target audience of this web application are patients, and necessarily these patients are usually above the age of 30. Most patients as such would not be particularly familiar with online patient appointment systems, and it is not rational to create something complex and difficult to use because that would only result in patients not using the Wintan Hospital's online patient appointment system.

Having complex designs will push our target audience away from the online patient appointment system and simply rely on the legacy/traditional patient appointment system that relies on papers and calls in order to get an appointment.

The Nawaloka Hospital patient appointment system, despite being one of the most famous private hospitals in the Negombo area is barely ever used. The estimated traffic according to a simple traffic sniffing test was somewhere around 1 to 5 per hour, which is very low and is not efficient.

The simple and easy to use design will attract people to using the patient appointment system. The best and only way to improve the user experience is to provide the majority of the target audience that has a significant boost in user-friendliness than the traditional call/paper based appointment system.

Which is why, the simple design, easy appealing colors and less content are the core of this web application.

TECH STACK

As this is a full-stack web application, the stack I choose will be split into frontend and backend this is the stack I will be using and the justification as to why I used said technologies.

Frontend Stack:

- HTML – For basic semantic markup and layouts
- CSS – For styling the layouts and semantic markup
- Javascript – To control the logic and the interactions of the layouts
- Bootstrap – To create a responsive navigation bar
- JQuery – As Bootstrap requires it

Backend stack:

- PHP – To handle dynamic content of the web application in a synchronous manner
- MySQL – A web-enabled RDBMS that require a server, to store data.

USER DOCUMENTATION

The user documentation will describe, explain and provide evidence for certain implications and the satisfaction of user and system requirements. Each page that exists within the web application will be promptly explained, every component available will be described as well.

Critical criticism, justification and comparisons will be made to the design documentation, so please read every word of each section.

Homepage

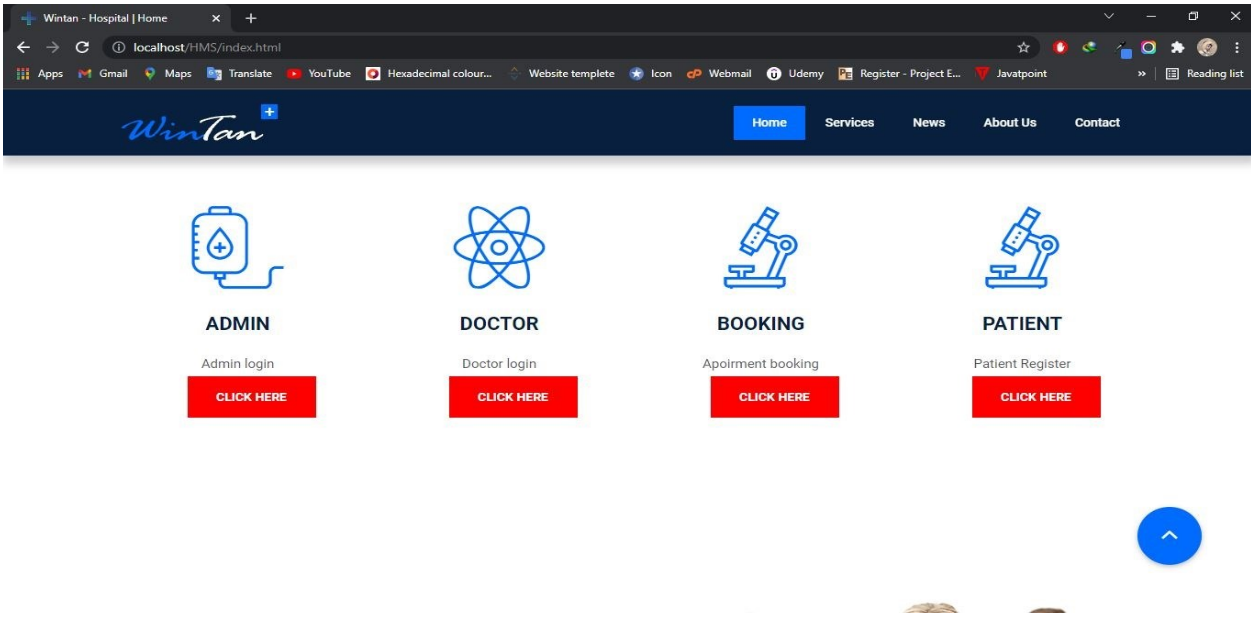


Figure 7: homepage wintan hospital

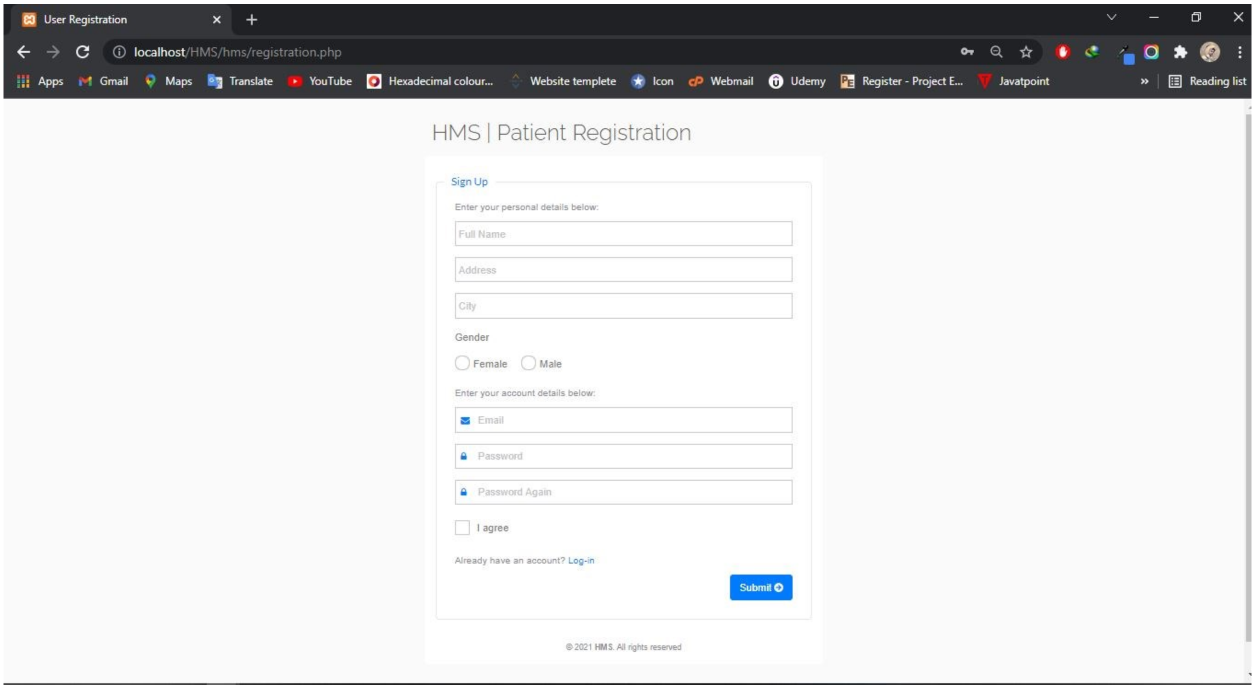
The first page you see when you visit a website is the home page. It welcomes every patient with a verse and picture confirming that Wintan Hospital is the hospital for their medical needs. As well as the section on Login and Registration, you will find information about wintan Hospital.

Compare, contrast and evaluation of design to development

The wireframe design of the homepage is almost identical to the implemented homepage, there is an image in the background and a greeting text. There is also a navigation bar that holds the Home, Service, News, Aboutus and contact with the brand of the hospital on the left hand side corner of the navigation bar.

The colors that were used however are different. The login and registration button will lead to the login and register webpage.

Register



HMS | Patient Registration

Sign Up

Enter your personal details below:

Full Name

Address

City

Gender

○ Female ○ Male

Enter your account details below:

Email

Password

Password Again

□ I agree

Already have an account? Log-in

© 2021 HMS. All rights reserved

Figure 8: Registration page wintan hospital

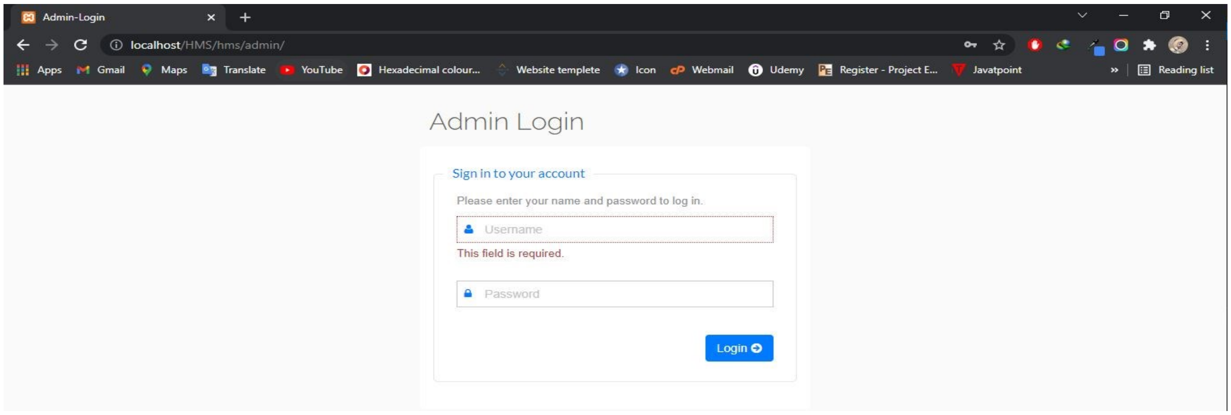
The register page is obtained when you click the submit button, it displays a form that will register a patient.

Compare, contrast and evaluation of design to development

The wireframe has any content on the form that will be displayed in the register page,

The register form requests information data that will be sent to the backend through an HTTP POST request.

Login



Admin Login

Sign in to your account

Please enter your name and password to log in.

Username

This field is required.

Password

Figure 9: Login page Wintan hospital

The login page is obtained when you click on the login button on the navigation bar. It too just like the register page, displays a form that has details to be filled with.

Compare, contrast and evaluation of design to development

The wireframe has any content on the form that will be displayed in the register page, after a user has registered, they may use their email ID and use the password they registered with. However, though upon registration, till you logout you will automatically be logged in.

Profile

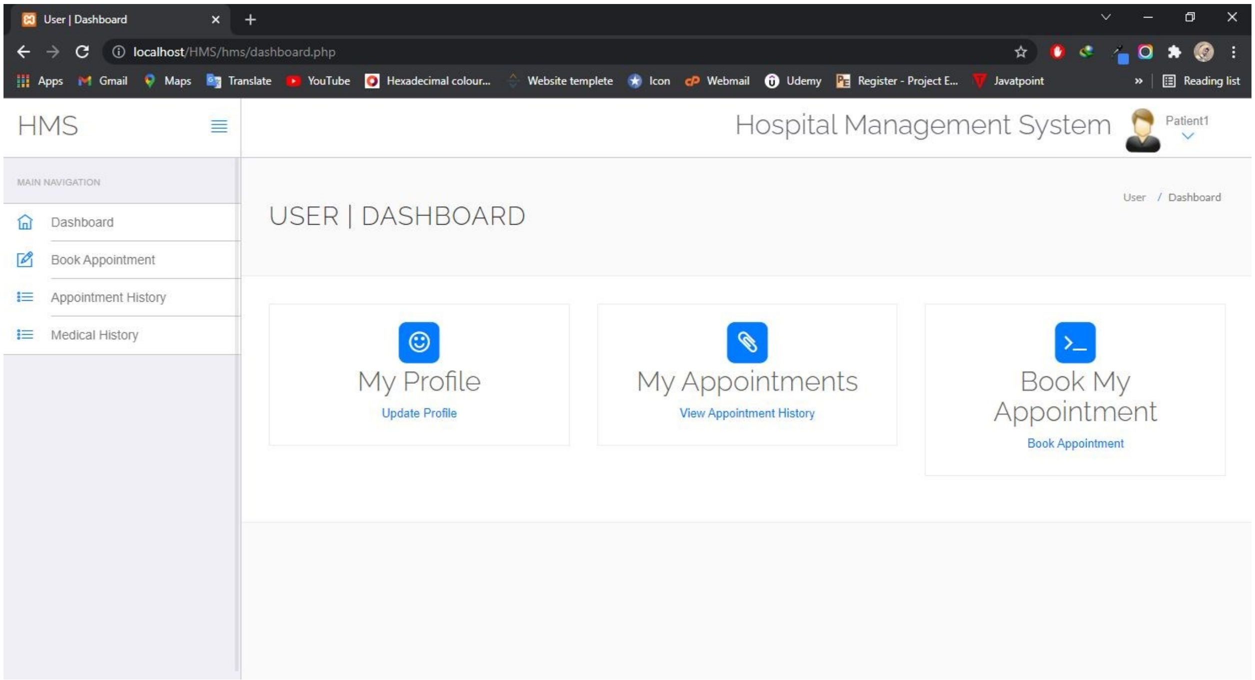


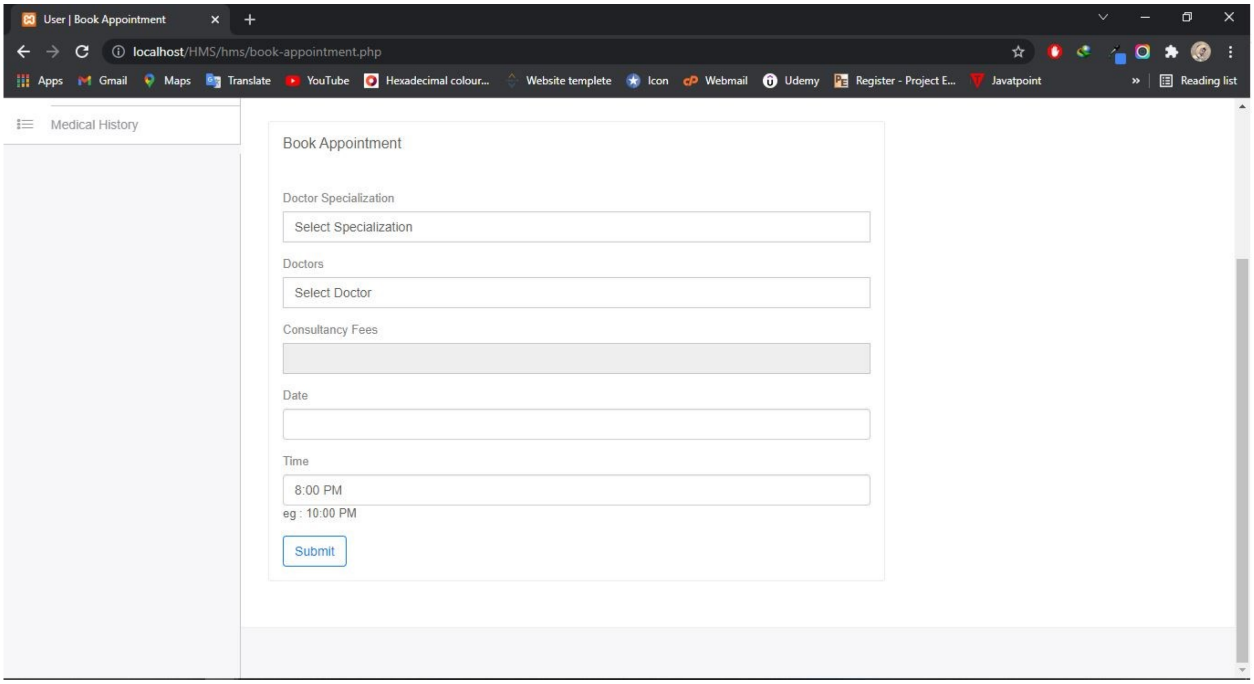
Figure 10: Profile page Wintan hospital

The profile page displays all the information of the logged in patient. The patient may only goto their profile after they have logged in, which means they must have registered in the wintan patient appointment system first.

Compare, contrast and evaluation of design to development

The profile page was not wireframed during the design stage as this was added as a utility page in order to boost user-friendliness and the overall user experience(UX) of the patient so that they can look at what data they have registered with. All the registered data except the password is listed here, in addition to the data that has been entered, the age field will be calculated at the backend and shown in the profile page.

Appoint a doctor



User | Book Appointment

localhost/HMS/hms/book-appointment.php

Medical History

Book Appointment

Doctor Specialization

Select Specialization

Doctors

Select Doctor

Consultancy Fees

Date

Time

8:00 PM

eg : 10:00 PM

Submit

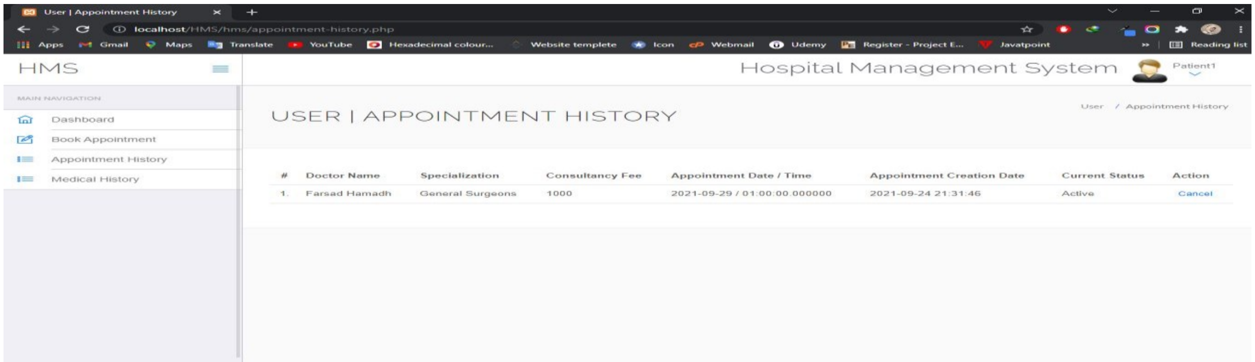
Figure 11: Appoint a doctor wintan hospital

The heart of the patient appointment system is the Appoint a Doctor page, this boasts a simple to use and less effort-based web application.

Compare, contrast and evaluation of design to development

The wireframe of the Appoint a doctor-page is entirely identical to the developed page, The dropdown in this form dynamically loads the doctors from the database that stores doctors information, the ID and the doctor name are combined here for processing easiness in other stages.

Your appointments



User | Appointment History

localhost/HMS/hms/appointment-history.php

HMS

Hospital Management System

Patient1

MAIN NAVIGATION

Dashboard

Book Appointment

Appointment History

Medical History

USER | APPOINTMENT HISTORY

User / Appointment History

#	Doctor Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Farsad Hamadh	General Surgeons	1000	2021-09-29 / 01:00:00.000000	2021-09-24 21:31:46	Active	Cancel

Figure 12: Appointments page wintan hospital

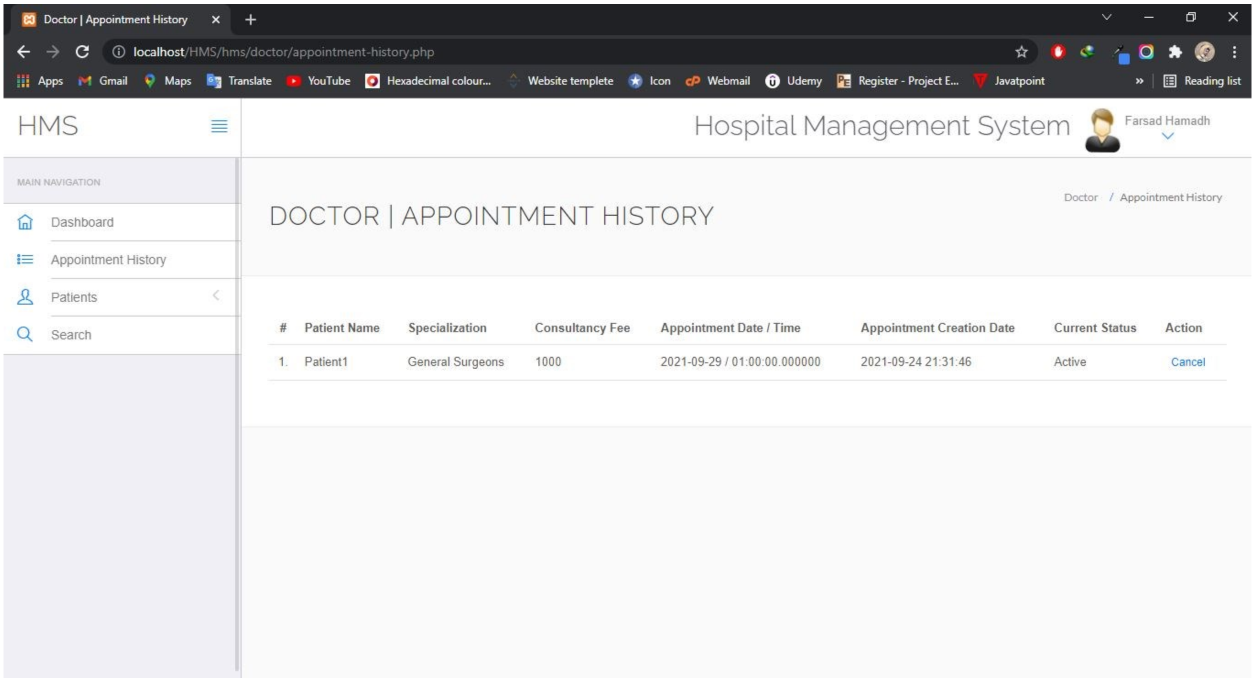
This page will show all the appointments that have been made, declaring verbosely the details of each appointment.

Compare, contrast and evaluation of design to development

Compare, contrast and evaluation of design to development

This page was not wireframed during the design stage and was added in the development stage in order to improve user experience and ease of use. It dynamically loads all the appointments made of a patient from the database. The Appointment IDs are dynamically created during the phase of creating an appointment, the doctor's information are taken from the database as well.

Doctor schedule



DOCTOR | APPOINTMENT HISTORY

#	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Patient1	General Surgeons	1000	2021-09-29 / 01:00:00.000000	2021-09-24 21:31:46	Active	Cancel

Figure 13: Doctor schedule page wintan hospital

This page will be accessible from the Appointments dropdown and is the last of the dropdown, it shows all the doctors available for appointment in the wintan hospital. Compare, contrast and evaluation of design to development

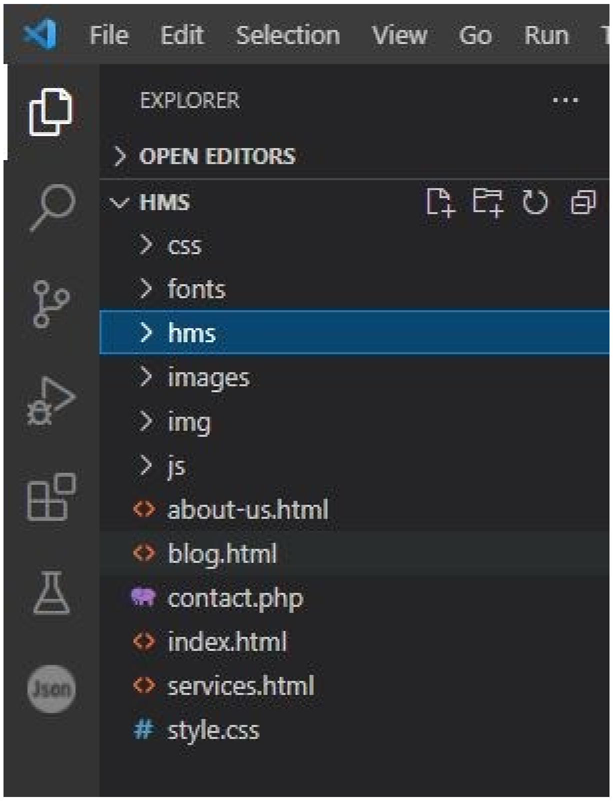
Compare, contrast and evaluation of design to development

This page was not wireframed during the design stage and was added in order to boost user experience and ease of use. All the values are loaded directly from the database and as such can only be added to the database.

EVIDENCE FOR LO3 and LO4

The project-tree overview

The DAOs are the modals, the views are the views and the routes are the controllers within the web application. The HMS folder contains all the static content like the css, images, javascript and html used. The hms is the backend contains all the data of the web application.



EXPLORER ...

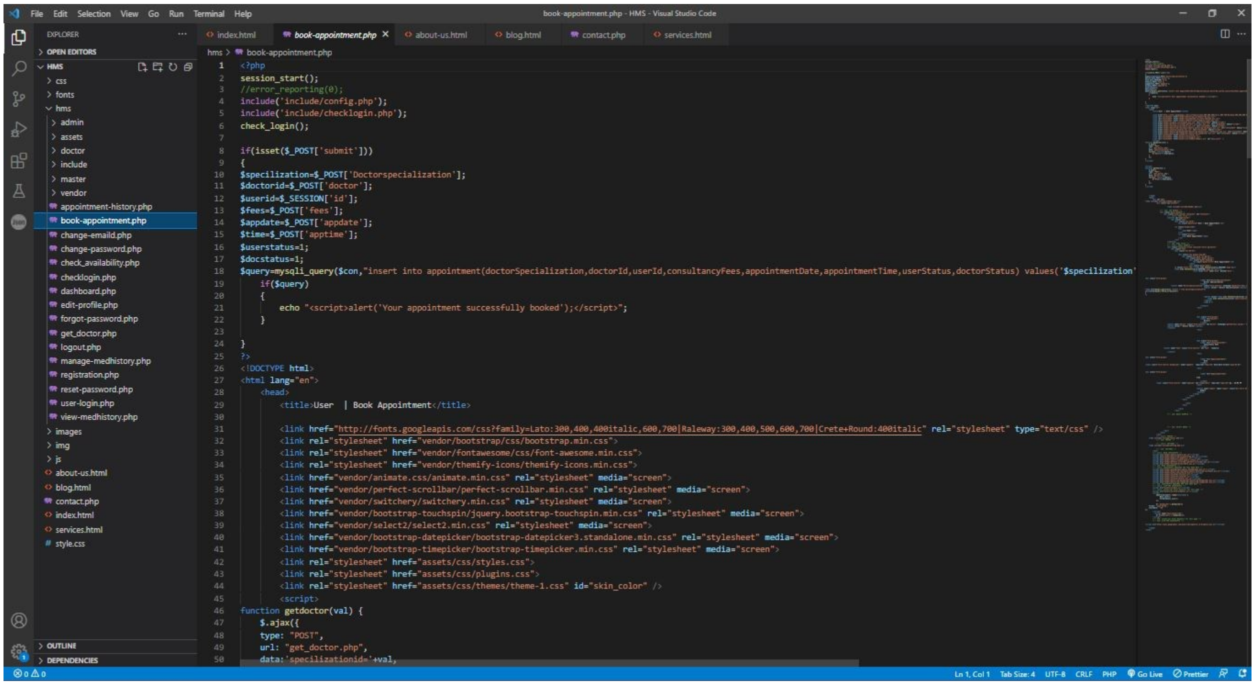
> OPEN EDITORS

∨ HMS
- > css
- > fonts
- > hms
- > images
- > img
- > js
- about-us.html
- blog.html
- contact.php
- index.html
- services.html
- style.css

Figure 14: project tree

Example of a view file (Using PHP, HTML and Javascript)

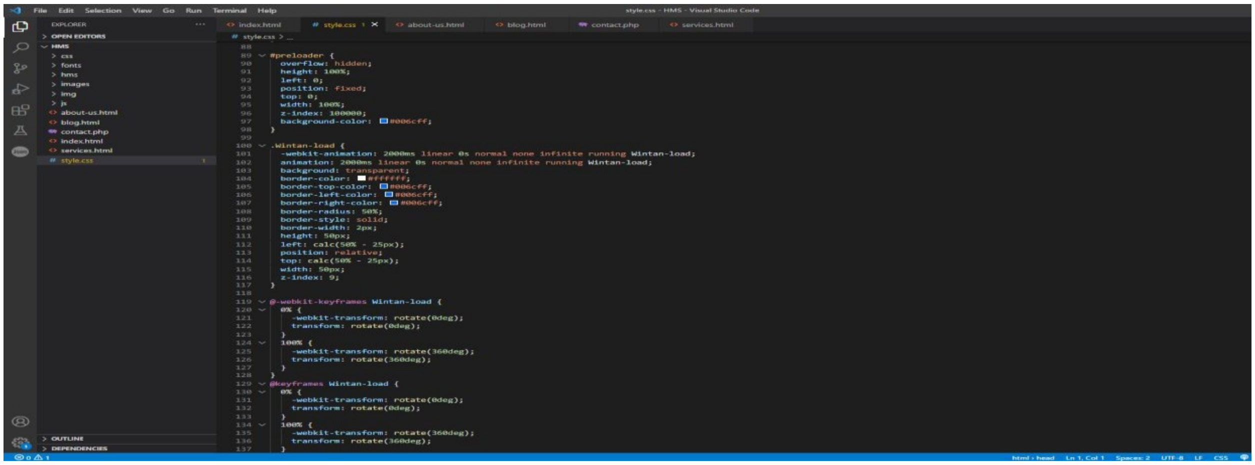
The following image is one of the many view components written using PHP, HTML and Javascript within the web application. The following is the Login-Form's code. The Javascript is used for form validation, to check if the fields are empty and then submit the form data.


book-appointment.php - HMS - Visual Studio Code
index.html
book-appointment.php
about-us.html
blog.html
contact.php
services.html
hms > book-appointment.php
Ln 1, Col 1 Tab Size: 4 UTF-8 CRLF PHP Go Live Prettier

```
<?php
session_start();
//error_reporting(0);
include('include/config.php');
include('include/checklogin.php');
check_login();

if(isset($_POST['submit']))
{
$specilization=$_POST['Doctorspecialization'];
$doctorid=$_POST['doctor'];
$userid=$_SESSION['id'];
$fees=$_POST['fees'];
$appdate=$_POST['appdate'];
$time=$_POST['apptime'];
$userstatus=1;
$docstatus=1;
$query=mysqli_query($con,"insert into appointment(doctorSpecialization,doctorId,userId,consultancyFees,appointmentDate,appointmentTime,userStatus,doctorStatus) values('$specilization'
    if($query)
    {
        echo "<script>alert('Your appointment successfully booked');</script>";
    }

}
?>
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>User  | Book Appointment</title>

        <link href="http://fonts.googleapis.com/css?family=Lato:300,400,400italic,600,700|Raleway:300,400,500,600,700|Crete+Round:400italic" rel="stylesheet" type="text/css" />
        <link rel="stylesheet" href="vendor/bootstrap/css/bootstrap.min.css">
        <link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.min.css">
        <link rel="stylesheet" href="vendor/themify-icons/themify-icons.min.css">
        <link href="vendor/animate.css/animate.min.css" rel="stylesheet" media="screen">
        <link href="vendor/perfect-scrollbar/perfect-scrollbar.min.css" rel="stylesheet" media="screen">
        <link href="vendor/switchery/switchery.min.css" rel="stylesheet" media="screen">
        <link href="vendor/bootstrap-touchspin/jquery.bootstrap-touchspin.min.css" rel="stylesheet" media="screen">
        <link href="vendor/select2/select2.min.css" rel="stylesheet" media="screen">
        <link href="vendor/bootstrap-datepicker/bootstrap-datepicker3.standalone.min.css" rel="stylesheet" media="screen">
        <link href="vendor/bootstrap-timepicker/bootstrap-timepicker.min.css" rel="stylesheet" media="screen">
        <link rel="stylesheet" href="assets/css/styles.css">
        <link rel="stylesheet" href="assets/css/plugins.css">
        <link rel="stylesheet" href="assets/css/themes/theme-1.css" id="skin_color" />
        <script>
function getdoctor(val) {
    $.ajax({
    type: "POST",
    url: "get_doctor.php",
    data:'specilizationid='+val,
```

Figure 15: Using PHP, HTML and javascript evidence

Example of a style file (Using CSS)

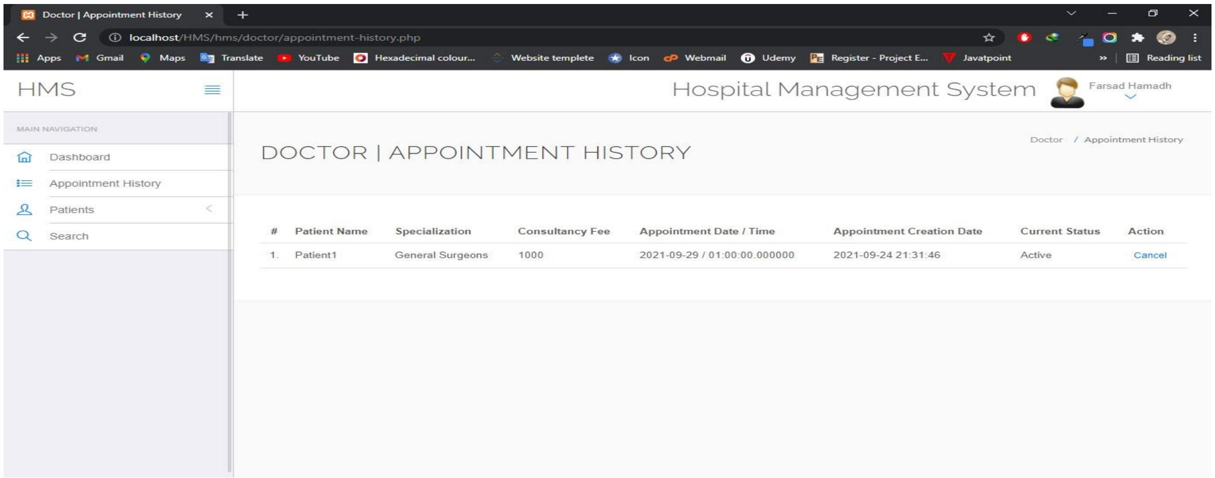


```

#preloader {
  overflow: hidden;
  height: 100%;
  left: 0;
  position: fixed;
  top: 0;
  width: 100%;
  z-index: 100000;
  background-color: #006cff;
}

.Wintan-load {
  -webkit-animation: 2000ms linear 0s normal none infinite running Wintan-load;
  animation: 2000ms linear 0s normal none infinite running Wintan-load;
  background: transparent;
  border-color: #ffffff;
  border-top-color: #006cff;
  border-left-color: #006cff;
  border-right-color: #006cff;
  border-radius: 50%;
  border-style: solid;
  border-width: 2px;
  height: 50px;
  left: calc(50% - 25px);
  position: relative;
  top: calc(50% - 25px);
  width: 50px;
  z-index: 9;
}

@-webkit-keyframes Wintan-load {
  0% {
    -webkit-transform: rotate(0deg);
    transform: rotate(0deg);
  }
  100% {
    -webkit-transform: rotate(360deg);
    transform: rotate(360deg);
  }
}
@keyframes Wintan-load {
  0% {
    -webkit-transform: rotate(0deg);
    transform: rotate(0deg);
  }
  100% {
    -webkit-transform: rotate(360deg);
    transform: rotate(360deg);
  }
```

Figure16: style.css evidence

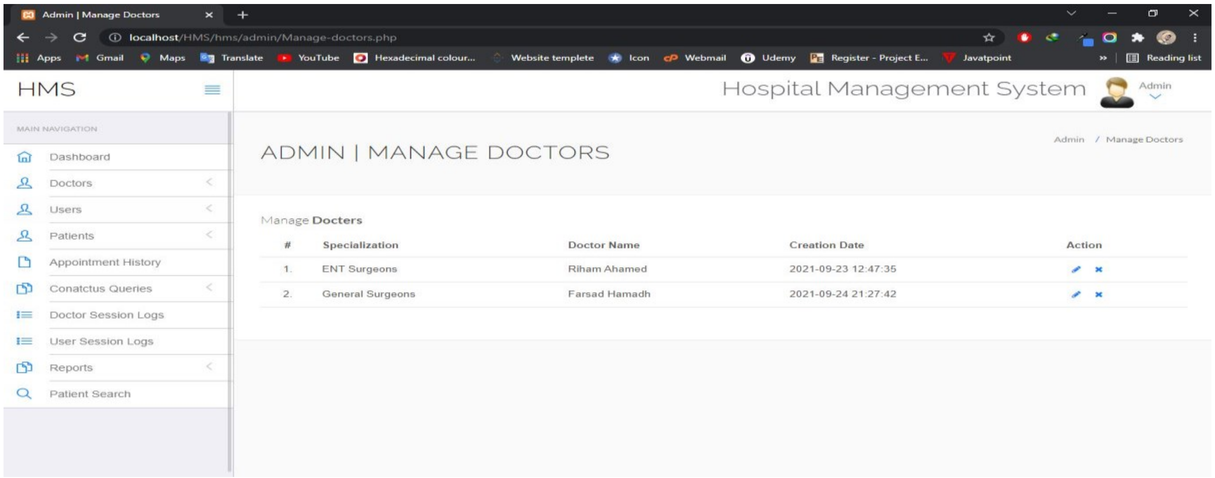
The following image is one of the many view components written using css within the web application. The following is the style.css code. The css is used for form style.

Other Required Evidence



DOCTOR | APPOINTMENT HISTORY

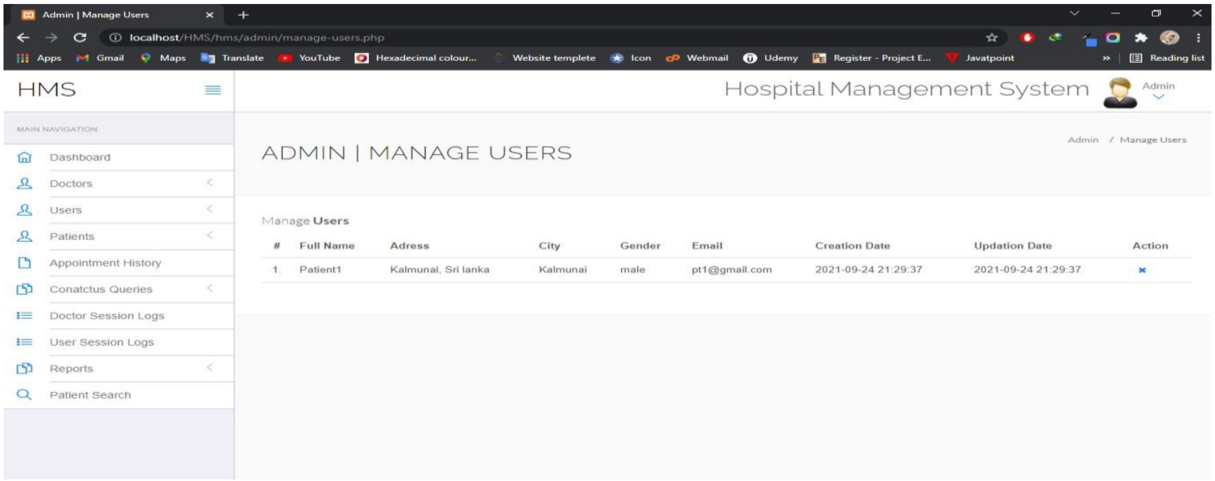
#	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Patient1	General Surgeons	1000	2021-09-29 / 01:00:00.000000	2021-09-24 21:31:46	Active	Cancel



ADMIN | MANAGE DOCTORS

Manage **Docters**

#	Specialization	Doctor Name	Creation Date	Action
1.	ENT Surgeons	Riham Ahamed	2021-09-23 12:47:35	✎ ✖
2.	General Surgeons	Farsad Hamadh	2021-09-24 21:27:42	✎ ✖



ADMIN | MANAGE USERS

Manage **Users**

#	Full Name	Adress	City	Gender	Email	Creation Date	Updation Date	Action
1.	Patient1	Kalmunai, Sri lanka	Kalmunai	male	pt1@gmail.com	2021-09-24 21:29:37	2021-09-24 21:29:37	✖

Figure 15: Other database data view evidence

TECHNICAL DIFFICULTIES FACED DURING THE DEVELOPMENT

1. Managing the state of the web application, in order to identify which user is logged in
2. Performance considerations, so that the UI does not look bloated or does not take too long to load
3. Mobile-first design, due to using cards it was difficult to make them look good on mobile screens, however it was achieved using CSS3 media queries and CSS3 flexboxes.
4. Overall memory efficiency, Javascript is known to be sloppy when it comes to maintaining the memory, there were a few memory leaks that were created during the development.
5. Troubles with walking the routes directory to mount the each router controller to the web application.
6. Loading data from the database without making the frontend lag

TEST SUITE

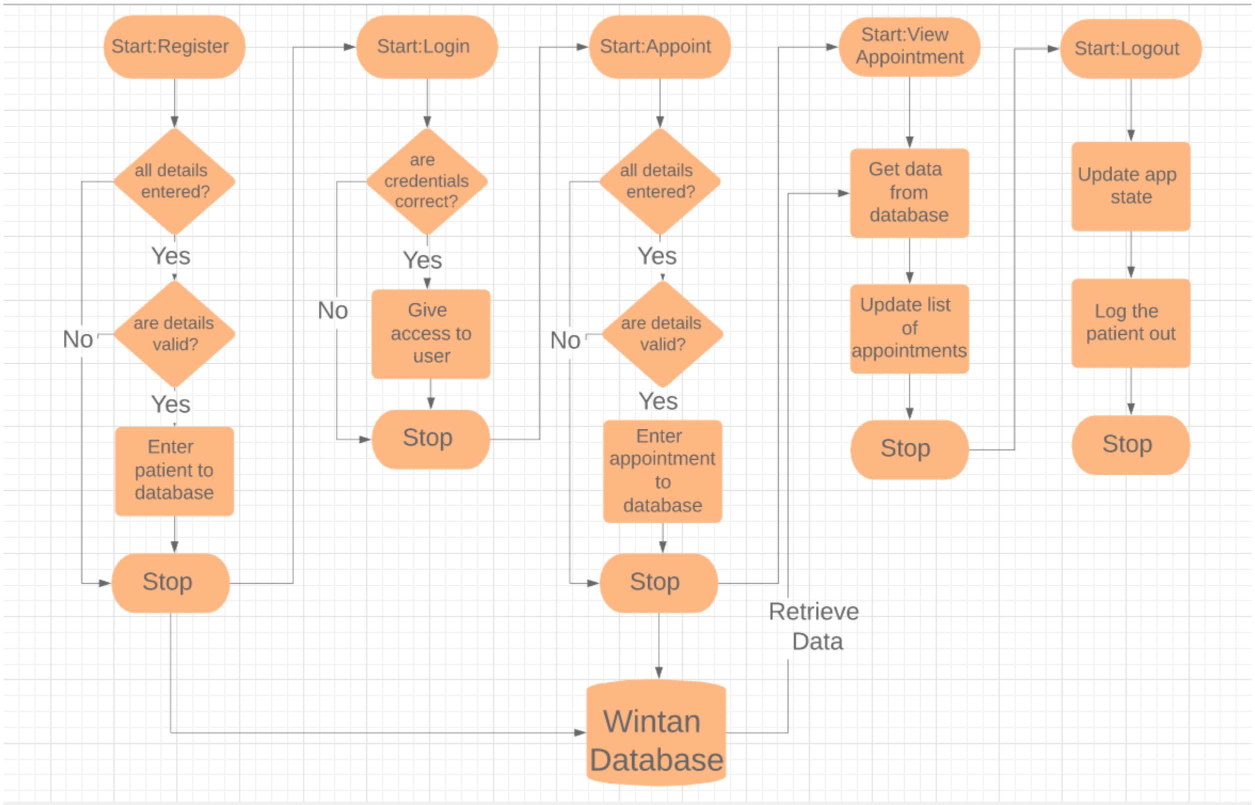


Figure 16: Date process

Figure 18, shows the overview of the test suite that will be used to test the full stack application. During this test suite, the following things will be done:

- Chrome DevTools performance profiler will be used to analyze the performance of the web application.
- Chrome DevTools memory snapshot system will be used to analyze how much memory is being consumed and provide deeper analytics on where memory is being used.
- Expected results will be matched to the actual results.

Table 1: Test report table

Test Name	**Expected**	**Actual**	**Comments**
Register a patient wrong detail	Patient is don't add to the database	Patient is don't add to the database	The test has been passed
Register a patient correct detail	Patient is added to the database	Patient is added to the database	The test has been passed
Login as a patient	Redirect patient dashboard page	Redirect patient dashboard page	The test has been passed
Add an appointment wrong detail	Your appointments don't add to the database	Your appointments don't add to the database	The test has been passed
Add an appointment correct detail	Your appointments added to the database	Your appointments added to the database	The test has been passed
Logout	User should be redirected to home page	User gets redirected to the home page	The test has been passed

Performance Test

The performance test shows that 24466 milliseconds were taken for the entire test.

registering a patient -> logging in -> making an appointment -> viewing it -> logging out.

24466 milliseconds translates to 24.5 seconds approximately, for 5 actions performed it is fair to say that the wintan hospital appointment system is very fast.

As you see, most of the time is spent idle which means the time I spent on inputting data which accounts for roughly 22.6 seconds. As such during the whole test suite, the time it took for loading, scripting, rendering and painting approximately took only 1.9 seconds!

This is blazing fast for a full stack application and I'd say that this project is a complete success, in terms of performance

QUALITY ASSURANCE

There were 4 main features that allowed for maintaining and ensuring that quality assurance was a large factor that played in both the design stages and the development stages which are:

Stress/Load based testing – The performance test and the memory snapshots were taken to ensure that the web application will run fast and smoothly on all devices across any browser.

Simple testing – The test suite that checked things like registering an account, logging in and logging out were to ensure that all the features that cover the user requirements were functioning to the highest levels and there is no problem at all in each feature.

Wireframing during design stage – This allowed me to have a rough idea of how the backend architecture would be structured and as such resulted in a neat project tree.

Using technologies that have survived the test of time – Using technologies like bootstrap, PHP etc. that have long withstood the pressure of industrial demands and the challenges that each technology had to face through time made it very extensively suitable for this application, using them kept me from falling to common pitfalls that would've been met if I implemented certain things from scratch.

Using mobile-first design and development – This ensures that across all platforms and devices, this application will be usable and would not look unresponsive or difficult to interact with.

OVERALL SUCCESS REVIEW AND POSSIBLE IMPROVEMENTS

I would say that the Wintan Hospital Online Patient Appointment system came out extremely well and exceeded both my expectations and the expectations of my test suite.

- The reasons I say the project was successful are:
- All user and system requirements were met ideally
- Performance of the web application is perfect
- Memory used is very less
- Looks great on all devices from mobiles to tablets to laptops to 4K screens
- There are no notable errors or missing functionality that may annoy a user
- The application is very secure and handles security neatly
- The test suite came out saying that the application works perfectly

However, though there are some considerable improvements that can be made to the existing online appointments system, which are:

- Dropping PHP from the backend stack as PHP is obsolete, legacy and uses synchronous programming, which is quite slow
- Use sessions instead of having a global application state which would allow for easier maintainenance of the state as a global state is susceptible to change from anywhere
- Using media queries to load smaller sizes of the image for smaller screens, which would improve load time
- Have Search Engine Optimization so the website is possible to be found from search engines
- Compress all images used so it would improve the load time of the web application
- Use a bundler like Webpack which would allow to create minified Javascript files which would improve load time
- Use a CSS preprocessor to reduce the number of styling rules in the stylesheet which would result in faster loading