

Homeland - A project by Aventure Solutions

by

Ashiqur Rahman
19241011

An internship report submitted to the Department of Computer Science and
Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2021

© 2021. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Ashiqur

Ashiqur Rahman

19241011

Approval

The project titled “Homeland - A Project by Aveneur Solutions” submitted by

1. Ashiqur Rahman (19241011)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 26, 2021.

Examining Committee:

Primary Supervisor:
(Member)



Zulker Nien
Business Analyst
Inside Maps

Secondary Supervisor:
(Member)



Afrina Khatun
Lecturer
Department of Information and Communication Technology
Bangladesh University of Professionals

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

When signing a contract at Aveneur Solutions, I have also signed a Non Disclosure Agreement contract, therefore I was not able to include some of the information in my report.

Abstract

Avenueur Solutions is an aesthetic-first web design, digital content, and software development company - serving businesses both large and small. It is a creative IT company that provides a diverse set of services tailored to your specific needs. At the moment I am working on a project for Homeland - a company that aims to provide a modern residential society to its consumers at very low rates. We are making them an e-commerce website with React Js library for the frontend part, and ASP Dot Net Framework for the backend part. Their website is mainly to showcase their available flats, but it will have the added feature to book flats with a down payment, therefore we are integrating the SSLCOMMERZ payment platform into the site. We hope the project will be a success.

The report is broadly categorized into eight different chapters. In the introductory chapter, I talk briefly about both Avenueur Solutions and the project I am involved in. I have also talked about my reason for doing Internship at Avenueur Solutions and why I would benefit from it. In the second chapter, I have talked more extensively about both the project and the company culture at Avenueur Solutions. In the third chapter, I have talked about my training period in Avenueur Solutions, what languages and frameworks I have learnt. In the fourth chapter I have more extensively about the project details for my main project that is Homeland, and also the other projects that I have been involved in. In the fifth chapter, I have talked about the project teams that were made for each project, team events, and how we celebrate after a job well done. In the sixth chapter, I have talked about the challenges I have faced, and how I overcame them to grow both professionally and technically. In the seventh chapter I have assessed the internship program and have suggested improvements from both Avenueur's side and BRAC University's side. In the last chapter, I have written the conclusion to this report.

Acknowledgement

Firstly, all praise to the Great Allah for whom my internship has been completed without any major interruption.

Secondly, to my supervisor at Avenueur Solutions, Mr. Zulker Nien, for his kind support throughout the whole internship.

Thirdly, Afrina Khatun ma'am guided me and taught me how to write a proper report.

Fourthly, to Dr. Md. Golam Rabiul Alam, for helping me in my time of need.

And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iii
Abstract	iv
Dedication	iv
Acknowledgment	iv
Table of Contents	v
List of Figures	vii
1 Introduction	1
1.1 Origin of the Report	1
1.2 Reason for choosing Internship	1
1.3 Reason for choosing Aveneur Solutions	2
1.4 Data Collection Method	2
2 Project Information and Company Culture	3
2.1 Company Culture	3
2.1.1 Company Profile	3
2.1.2 Recruitment Process	4
2.1.3 Company Size and Branches	4
2.1.4 Software Development Architecture	4
2.1.5 Software Development Practice	5
2.1.6 Projects Currently being handled by Aveneur Solutions	5
2.2 Project Information	6
2.2.1 Background of the Project	6
2.2.2 Project Scope	6
2.2.3 Project Limitations	7
3 Starting Training	8
3.1 Introduction to React JS	8
3.2 Introduction to React Semantic UI	9
3.3 Introduction to ASP .Net Core and APIs	9

4	Project Details and Tools Used	11
4.1	Projects I Have Worked On	11
4.1.1	Portfolio Website for Aveneur	12
4.1.2	Portfolio Website for Senoark	13
4.1.3	Your Lawyer App for LegalX	17
4.1.4	Portfolio Website for Shawn Novel & Associates	21
4.2	Main Project I Have Worked On - Homeland	24
4.2.1	Project Description	24
4.2.2	Project Breakdown Structure	25
4.2.3	Job Responsibilities	26
4.2.4	Functional Requirements	27
4.2.5	Non-Functional Requirements	28
4.2.6	Time Distribution	29
4.3	Tools used	29
5	Project Teams	32
5.1	Team Assembly	32
5.2	Team Building Exercises	32
5.3	Team Events & Celebration	33
6	Professional and Technical Growth	34
6.1	Professional Growth	34
6.1.1	Adaptability To The Office Environment	34
6.1.2	Adaptability To Remote Work Environment	34
6.1.3	Punctuality	35
6.1.4	Maintaining Deadlines	35
6.1.5	Communication Skills	35
6.2	Technical Growth	36
6.2.1	JavaScript Proficiency	37
6.2.2	Framework Proficiency	37
6.2.3	Different Approaches To A Problem	38
7	Self-Assessment Suggested Improvements	40
7.1	Self Assessment of Current Internship Program	40
7.2	Suggested Improvements for future internship programs	40
8	Conclusion	42
	Bibliography	44

List of Figures

2.1	MVC Architecture	4
2.2	Agile Methodology	5
3.1	API Logic	9
4.1	Avenueur Solutions Homepage	12
4.2	Avenueur Solutions About Us	13
4.3	Avenueur Solutions Connect	13
4.4	Senoark Startup Incubator	14
4.5	Senoark Concerns	15
4.6	Senoark Admin Panel	15
4.7	Senoark & Time Distribution	16
4.8	Your Lawyer App Screens	17
4.9	Your Lawyer Admin Panel	18
4.10	Your Lawyer App Time Distribution	20
4.11	Shawn Novel & Associates Homepage	21
4.12	Shawn Novel & Associates About Us	21
4.13	Shawn Novel & Associates Admin Panel	22
4.14	Shawn Novel & Associates Time Distribution	23
4.15	Homeland Homepage	24
4.16	Homeland Admin Panel	25
4.17	Project Breakdown Structure	25
4.18	Homeland Profile Page	26
4.19	Homeland Flats Page	27
4.20	Homeland Time Distribution	29
5.1	Senoark Football Cup	33
6.1	Successful Handover	36
6.2	JavaScript Array.map()	37
6.3	Request Functions	38
6.4	Login Function	39
6.5	Messy Login Function	39

Chapter 1

Introduction

My journey in Aveneur is coming to an end, as I have been here for more than 6 months, but I have gotten the opportunity to learn a lot and work on many production-level projects. I have learned how to work in a team well and how to adapt to different roles. I had achieved both professional and technical growth.

In today's world, everything is connected to a huge network that we call the Internet. As a result, Businesses, Education Institutes, etc. all over the world, be it big or small, are being digitized. All the services they used to provide are now being provided through web applications.

Taking traditional e-commerce websites in consideration, we at Aveneur Solutions have made a modern, well optimized and user-friendly e-commerce website for our client Homeland. This website is an online flat booking platform for the flats inside the Homeland Residential Society located near Gazipur. Users can book flats with a down payment by paying through the payment gateway integrated into the website. When they decide to buy the flat, they pay the remaining amount in cash. As a whole, the system is very robust and easy to use.

1.1 Origin of the Report

This report originates from the CSE Degree requirements from BRAC University. As a prerequisite for completing the CSE degree, the students must either complete an internship program, write a thesis paper, or make a project. In an internship, a student must stay employed in a company for at least 6 months, and within these 6 months a report has to be written with the contents of the participant's learnings and activities performed within the company.

1.2 Reason for choosing Internship

I have chosen to do internship rather than writing a thesis or project because it will help my career more if I have firsthand experience working in a company. I expect to learn more about the IT industry and how it operates. From programming perspective, I expect to learn different frameworks, architectural patterns, clean and better ways of writing code so that I can grow my skills as a developer further, and

hopefully use it for future jobs. From system analysis perspective, I expect to learn how to analyze requirements from a client's needs, look at how project planning is performed, and how the project can be financially broken down. I also would learn to adapt to the software development methodology commonly followed by the company, and how to work alongside a team of developers.

1.3 Reason for choosing Aveneur Solutions

The reason I have chosen Aveneur Solutions is because it is a relatively small and new company, therefore it follows a startup culture. In this culture, I would have to directly collaborate with my higher ups and even some of the clients. Since it is a small company, I will learn how to adapt to change and uncertainty as my job responsibilities might have a chance of shifting. As companies like this have limited resources, every employee working here would be working here out of passion and not because they will have high salaries. Working around these people will definitely have a positive impact on my attitude for working in the industry.

1.4 Data Collection Method

There are two types of data collection methods: primary and secondary. Primary data collection refers to first hand data gathered by the researcher himself and Secondary data collection refers to data collected by someone else earlier. The accuracy and reliability of primary data is always more than secondary data, but secondary data can be collected in a short amount of time as compared to primary data. Primary data collection is also more expensive and harder to acquire, as the researcher has to be very involved with the research and be on site most of the time. As for secondary data, it can be gathered from the internet, so it is very inexpensive and easy to collect.

In this report I have not used secondary data collection methods at all. Since I have joined the company when it was still in its starting stage, it also would not have been possible for me to collect any secondary data. The primary data collection methods that I have used include:

- Gathering data from the work that has been assigned to me.
- Gathering data from direct conversations to the developers and the project manager.
- Gathering data through direct observations.

Chapter 2

Project Information and Company Culture

2.1 Company Culture

In this section I have written about the company culture, like details about Aveneur, their recruitment process, software architecture and methodologies, etc.

2.1.1 Company Profile

Aveneur Solutions is an aesthetic-first web design, digital content and software development company - serving businesses both large and small. We are a creative IT company that provides a diverse set of services tailored to your specific needs. We also promote innovation, develop ideas into reality and provide opportunities for businesses to grow. Their main services are to set up and maintain business dashboards, data mining, and reporting. They also provide the full range of consultancy and implementation of Cloud Computing Architecture by designing highly fault-tolerant, scalable, secured, compliant, and cost-effective solution according to need. [1]

Mission

Aveneur Solutions mission is to provide the highest-grade, bespoke web, application, software, and technological solutions to its valued client base in an affordable and time-efficient manner. They provide services to all types of companies from solo bloggers to large enterprises. [1]

Vision

We are on a journey to expand and grow into a ubiquitously trusted name in the fields of digital design and technology. In time, Aveneur Solutions shall span the globe to provide its digital development and technology transfer services in major markets of the world over, such as the United Kingdom, Singapore and the United States. The core values of the company shall always be integrity and social responsibility. [1]

Objectives

Aveneur Solution's goal is their clients' success. Therefore, they focus on every detail

of the client’s needs and in return try to provide a solution that is high-performance and user friendly. They also focus on being very friendly with their employees while maintaining a respectful hierarchy. They also push employees to improve themselves and provide all the resources necessary for their growth.

2.1.2 Recruitment Process

The requirement for the internship was to already have knowledge in JavaScript and Object-Oriented Programming. They took CVs by email and looked at candidates’ projects in GitHub or GitLab. After the screening process, they selected a few candidates for the coding test online. For the online test they gave a set of JavaScript problems to test candidates’ knowledge on JavaScript ES6. After that they interviewed the candidates that have qualified, and based on the interview they made their final decision.

2.1.3 Company Size and Branches

Avenue Solutions is a sister concern of Senoark – a digital marketing agency. By itself, Avenue consists of 11 people, 7 of which are developers including myself, 2 people on marketing, a project manager and the chairman of the company. Despite being a separate company, we work alongside Senoark in the same building and together we are 20 plus employees. Since Avenue is a small company, there is only one branch in Rampura, but they hope to expand very soon.

2.1.4 Software Development Architecture

The technology stack used in Avenue Solutions is ASP .Net Core, Microsoft SQL Server and React JS. This stack is based on the MVC pattern. The Model corresponds to data related logic like inserting, retrieving, updating and deleting information in the database. The View corresponds to the UI logic of the applications. Our view is made with React JS and it presents data to the user from the database. The Controller acts as an interface between Model and View components to process all the incoming requests, manipulate data using the Model component and interact with the Views to render the final output.

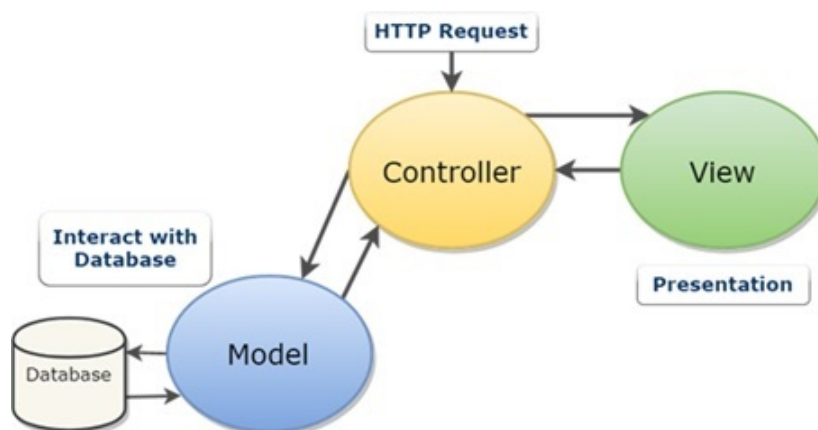


Figure 2.1: MVC Architecture

2.1.5 Software Development Practice

The software development practice which is commonly followed at Aveneur Solutions is Agile methodology. Agile is a process where a team breaks down a project into several stages, discovers requirements for those stages, constantly collaborates with the stakeholders and makes continuous improvements whilst making iterations of every stage. Since the product is delivered in small increments, the feedback gained from collaborations help with a lot of situations like reducing risk of making an inappropriate feature for clients, reduce future maintenance costs and making sure the clients get exactly what they need – resulting in customer satisfaction. It is also very easy to adapt to any changes required in the project and as the development team work together with the product owners, they will have more control over the project.



Figure 2.2: Agile Methodology

2.1.6 Projects Currently being handled by Aveneur Solutions

As Aveneur Solutions is a small company and is relatively new, the number of projects being developed is limited, but it is enough for the company to sustain and keep spreading its name by building a robust portfolio. Below are some of the projects being handled with brief descriptions:

- **Portfolio Website for Senoark** - Senoark is the mother company of Aveneur solutions, it is a digital marketing agency. The website has already been developed by our team is currently up and running at senoark.org. Right now, we have the job of maintaining it.
- **Portfolio Website for Aveneur** - Our own company website is hosted at aveneur.org
- **Your Lawyer App for LegalX** - Your Lawyer App helps people find lawyers in their district according to their areas of law. It is also an app where they can get free legal advice from the LegalX team. Both web application and mobile application are being made.

- **Portfolio Website for Shawn Novel & Associates** - Shawn Novel and Associates are another law firm which is looking to upgrade their existing website made by WordPress, into a fully customizable React website.
- **E-commerce Web Application for Homeland** - Homeland wants to make a platform where people can book flats within their own residential society.

2.2 Project Information

In this section I have written more about the project and its details like the project's objective and its background etc.

2.2.1 Background of the Project

The project I have chosen to write my report on, is the e-commerce web application for Homeland. Houses are the most secure investments people can make because it will not only guarantee a roof over their heads, it can also become a steady source of income through rent, but nowadays, it is very hard to find houses within a good society at an affordable price. Homeland offers a solution to this by making sure people experience modern societal benefits within a reasonable price with their own residential society. Homeland Residential Society is located near Gazipur and the price of the flats are more than enough reasonable to the middle-lower class of our society. It is primarily targeted towards people who work in garments factories and have to travel to Gazipur every day from very far away.

2.2.2 Project Scope

The website should portray a consolidated and steady image of Homeland. When users enter the site, they should readily know what the site is for just by looking at the landing page. It should be user friendly for their customers and at the same time be as eye-catching modern websites these days. The site should also be secure as it is an e-commerce site where payment is involved. Through the website admin panel, Homeland employees should be able to do the following:

- Add new flats and manage existing ones.
- Manage bookings made by their customers.
- Manage allotments made by their customers.
- See website traffic.
- Manage and see payment history of customers.

2.2.3 Project Limitations

Due to budget constraints and lack of man-power, this project was not created with high scalability in mind. Scalability in a web application refers to the ability to cope with increasing number of concurrent users in the site. Many problems in logic might persist due to a web application not being scalable. For example, in an e-commerce website, if two customers choose to buy a specific product, which has a stock quantity of one, at the same exact time, the system might produce a bug and confirm the order for both customers. These types of problems in scalability occurs when there are thousands of users using the site in a day, but since the website for Homeland is in a niche market catering to a specific group people, there is a very low possibility that it might come across such issues. However, if it does, and the issues are occurring more frequently than preferred, the whole internal architecture of the web application will need to be overhauled and the server will need to be changed according to the website traffic. As of now it is decided that it will be hosted in a local server with mediocre specifications.

One other limitation is that a lot of people in our country do not use smartphones, tablets or computers. The web application is not made keeping feature phones in mind, resulting in lower reach to customers.

Chapter 3

Starting Training

Avenueur Solutions were only taking in interns who had a good grasp on JavaScript and Object-Oriented Programming, therefore the training program consisted of learning JavaScript libraries and frameworks for frontend. At first, out lead front end designer Mr. Zulker Nein gave me small things to code in HTML and CSS, like containers, buttons and some layouts. When I was given a course to watch on Udemy labeled “Complete guide to building an app with .Net Core and React” by Neil Cummings. It was a detailed course on how to create a project from scratch using React TypeScript and .Net Core. I was only told to watch the React related videos, and I did not need to learn TypeScript separately as the instructor in the video was teaching how TypeScript works along the way as well. My primary source of learning was not confined to these videos only, the full stack developers Mr. Ragib Ibne Hossain and Mr. Ashikur Rahman has also helped me a lot while I was learning React as well. They were the ones who primarily taught me how APIs work.

3.1 Introduction to React JS

React is an open-source front-end JavaScript library for building user interfaces or UI components. It was created and is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of both SPA (Single Page Application) and Mobile Applications.

React is a declarative library which makes it easy to develop Interactive UIs. Parts of the site can be made into components and be re-used if needed. If react detects a change in a component, only that part is re-rendered instead of the whole page. This is done with the use of states. States preserve data inside and makes changes to the page if the data inside the state changes.

React Components take in and passes data through props from one component to another. Multiple props can be passed down to a component, and be used to mutate states, but keeping track of them might get hectic if the whole app is complex. Therefore, state management libraries are used to help keep track of the states. They consist of a centralized state that can be used in any components throughout the app. Mutations to that state would make mutations where ever the data is used. There are many state management libraries supported by react. We have used MobX as our state management library as it is easy to use quick to learn.

3.2 Introduction to React Semantic UI

Like Bootstrap CSS, react has a similar library where UI components are already pre-built and can be used while making changes to the design through props. There are a number of pre-built elements like buttons, containers, inputs, lists, loaders and also a number of layouts like grids, menus, tables, etc. which can save a lot of time and make our codes shorter and cleaner. These components are already made responsive. Therefore, we do not have to write extra code for it to be compatible in mobile view.

3.3 Introduction to ASP .Net Core and APIs

For backend, we have used the framework ASP .Net Core. It is an open-source framework developed by Microsoft, for building fast and secure web apps and services within .Net. Due to time constraints of the project, I could not fully learn ASP .Net Core in time, but I did learn the necessary things needed to integrate it with React JS. ASP .Net is cross-platform, meaning it can be run on Windows, Linux and MacOS. Using this will allow us to build services for web browsers, mobile apps, etc. These web services are called Web Application Programming Interfaces (APIs).

APIs act as the intermediary between two software, and allows them to communicate. An API makes requests to the system and based on those requests; the system generates the response for that API. A Web API defines endpoints and valid response, request formats. A web browser or mobile app can hit that endpoint and get a response back from it. The services made through APIs can vary from being web notifications or web storage. Different APIs provide different levels of security and privacy. Multiple APIs can be combined to make a composite API which provides a collection of services. There are multiple API architectures and protocols. For this project we have implemented REST API. Representational State Transfer

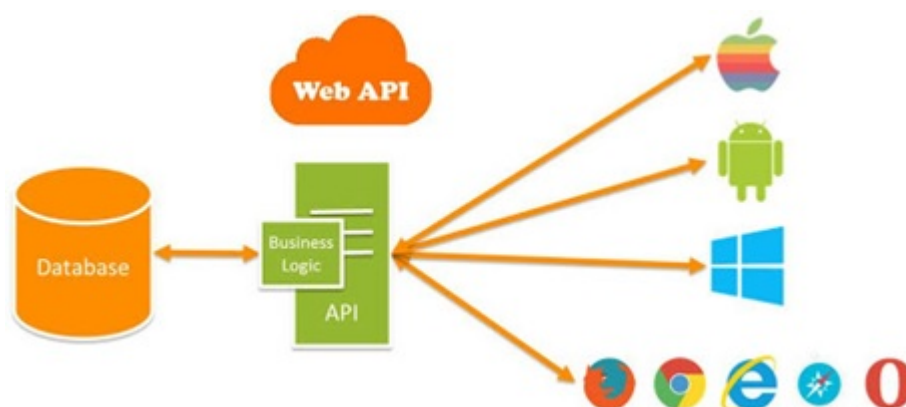


Figure 3.1: API Logic

(REST) is one of the most popular API frameworks for web development. In a REST API, the backend and frontend need to be separated, which is why our technology stack is more suited towards it. REST API works mainly around CRUD operations. CRUD stands for Create, Read, Update and Delete. These are based on 4 primitive

database operations: Create – INSERT, Retrieve – SELECT, Update – UPDATE and Delete – DELETE. In REST API, these are also mapped to request types from front end. Like Create – POST, Read – GET, Update – PUT/PATCH, Delete – DELETE. These requests determine the type of operation to be performed when an API is called from the backend. In REST API, GET requests are used to get data from the database so that it may be used in the frontend. POST requests take in data from the user and are used to create new entries into the database. PUT/PATCH requests are used to update exiting data in the database and DELETE requests are used to delete unwanted data in the database.

Chapter 4

Project Details and Tools Used

As I started my internship at Aveneur Solutions, at first, I was given small tasks like building a landing page or writing JavaScript functions, which were all designed to test my skills in programming. Later I was given work from official projects that the company had been working on. I was never fully assigned to one project as I was an intern, so I had the opportunity to work on various different projects before choosing my main project for the internship, which is Homeland. The Projects I have been working on are explained in detail, including their requirement analysis and time distribution per project.

The requirement analysis mainly consists of two parts – Functional Requirements and Non-functional Requirements.

Functional requirements define the basic system behavior. This includes what the system does or cannot do and can be thought of in terms of how the system responds to given inputs. Features that allow the system to function as it was intended. If the functional requirements are not met, the system will not work. Functional requirements are product features and focus on user requirements.

Non-functional requirements specify the quality attribute of a software system or a project. The software system is judged based on the responsiveness, usability, security, portability, and the other non-functional standards that are critical to the success of the software system.

The estimated time required to end a project successfully defines process-wise time distribution. This helps the developers create a mind map as to how efficiently they need to work in order to meet the deadlines. Time distribution is greatly needed to complete any project.

4.1 Projects I Have Worked On

In this section, other projects that I have worked on are described in detail, complete with screenshots and what part of the projects I have worked on. These are the projects that helped me learn how to code well enough to make me prepared for big projects like an ecommerce.

4.1.1 Portfolio Website for Aveneur

First and foremost, Aveneur need a website that looked as good as all the modern software company websites and it needed to clearly portray our mission, vision, goals and the types of services we provide. It needed to be fast and reliable as well. We initially thought of making the website using basic HTML, CSS and JavaScript, but we wanted to get rid of the long loader that existed in between screens. React loads up the all the CSS and JavaScript files all at once, therefore after the initial load, the other pages will load instantly. Since the initial loading time was higher than usual, we had to optimize the website by compressing images and refactoring code. The owners of the company wanted an option to post blogs, new job openings and edit site contents therefore to handle the backend, we have used ASP .Net framework. Below is a sample picture of the Aveneur Solutions Website: For the

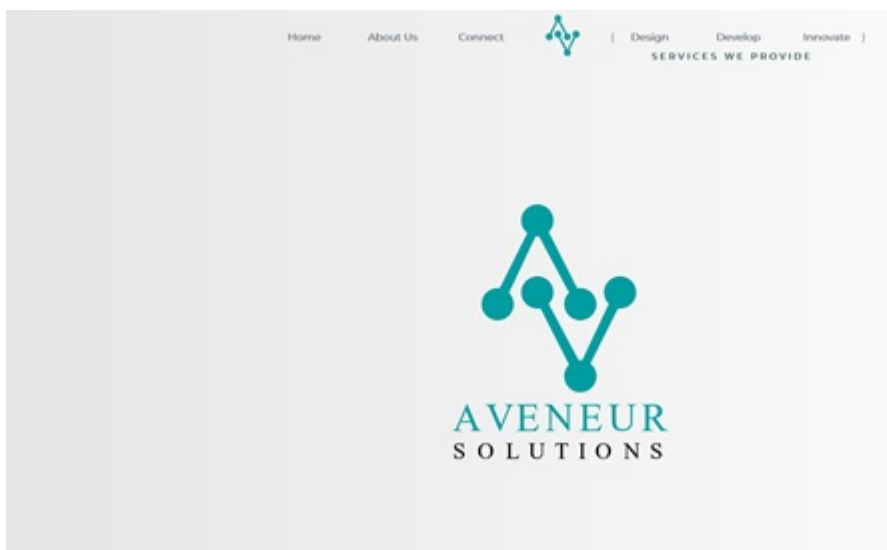


Figure 4.1: Aveneur Solutions Homepage

Aveneur website I have worked as a front-end developer and had made the About Us page and the connect page. For the About Us page, I have used the AOS – animate on scroll library, therefore the different components on about us are animated when the page is scrolled.

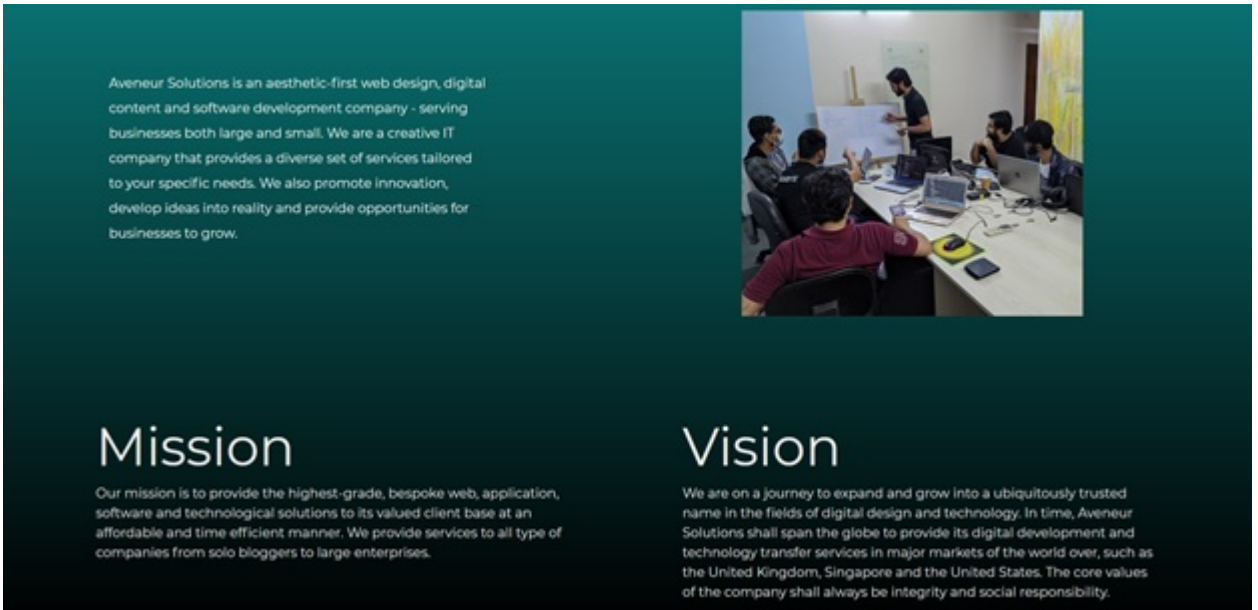


Figure 4.2: Avenueur Solutions About Us

The description on the left and the image on the right fade in from the left and right respectively. Mission and Vision are not visible in the beginning, but when the user scrolls, they both fade in from the bottom left and bottom right.

For the connect page, I also used the AOS library. For the google map location, I had placed it in an HTML iframe block. The logo, headline, contents fade in, and the map has a grow animation.

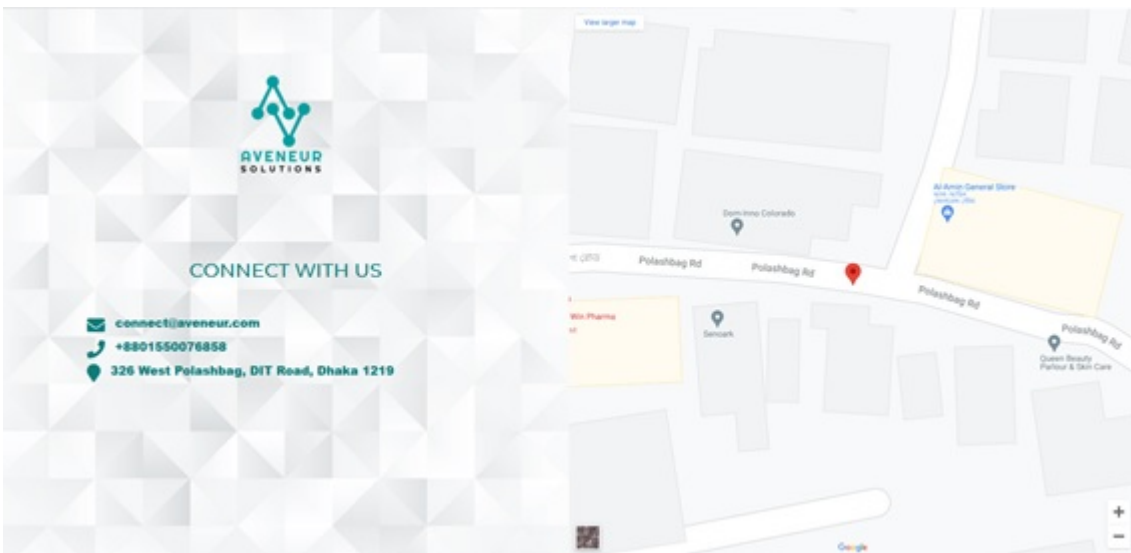


Figure 4.3: Avenueur Solutions Connect

4.1.2 Portfolio Website for Senoark

Senoark is the mother company of Avenueur solutions, it is a digital marketing agency. Like Avenueur, we have decided to go aesthetics first over the Senoark Website. Since,

it was modeled the same way as the Aveneur website, the first page has the option for customers to look at a glance, all of the main services being provided by Senoark which are – Startup Incubation and Digital Marketing Agency. Both of them have buttons which take the user to a page containing more detail. Startup Incubation is a place for users to kickstart their ideas by filling up the form shown below:

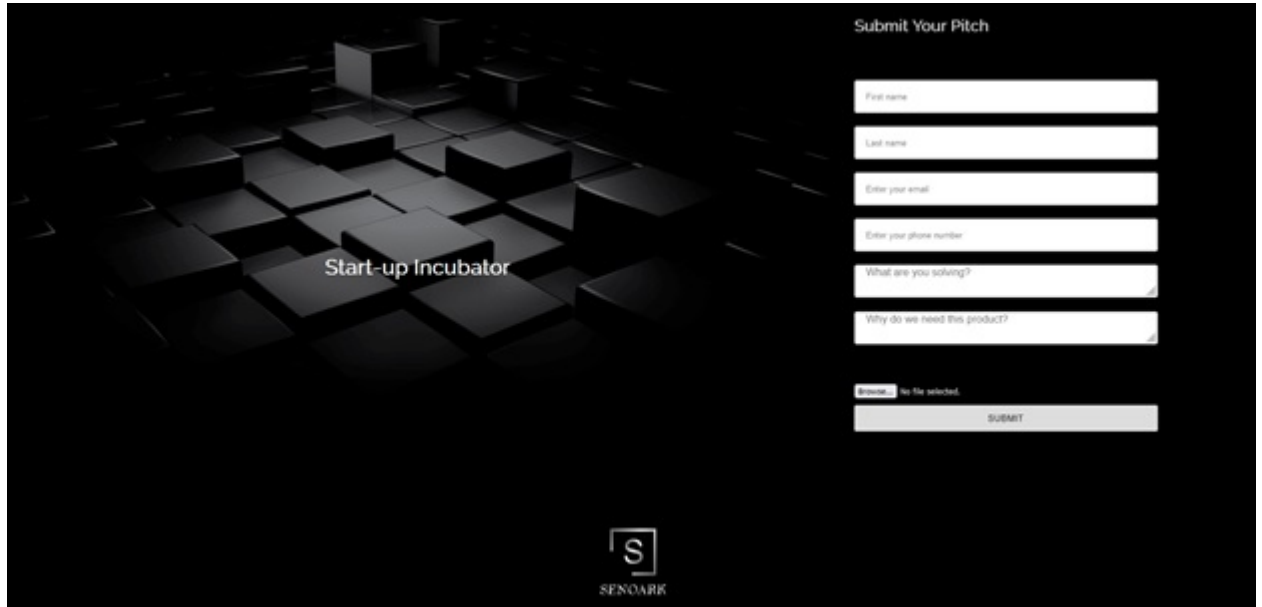


Figure 4.4: Senoark Startup Incubator

I had taken up the part of making the above page and the Concerns page. This page animated so that the image on the left takes up the whole screen first and the shrinks to the right. The form fields appear one by one, fading in from the left. For the form validation, I had used react hook form. When the submit button is pressed, an API call is made to the backend so that the data entered in the form is submitted to the database. The API call was made using axios.

As for the Concerns page, on page load, an API call is made to the backend to get the concern info which contain the id, image location, the button color, the button link and the content for a particular concern. There are no animations in this page as there are a lot of images and it might become slow to add more images.

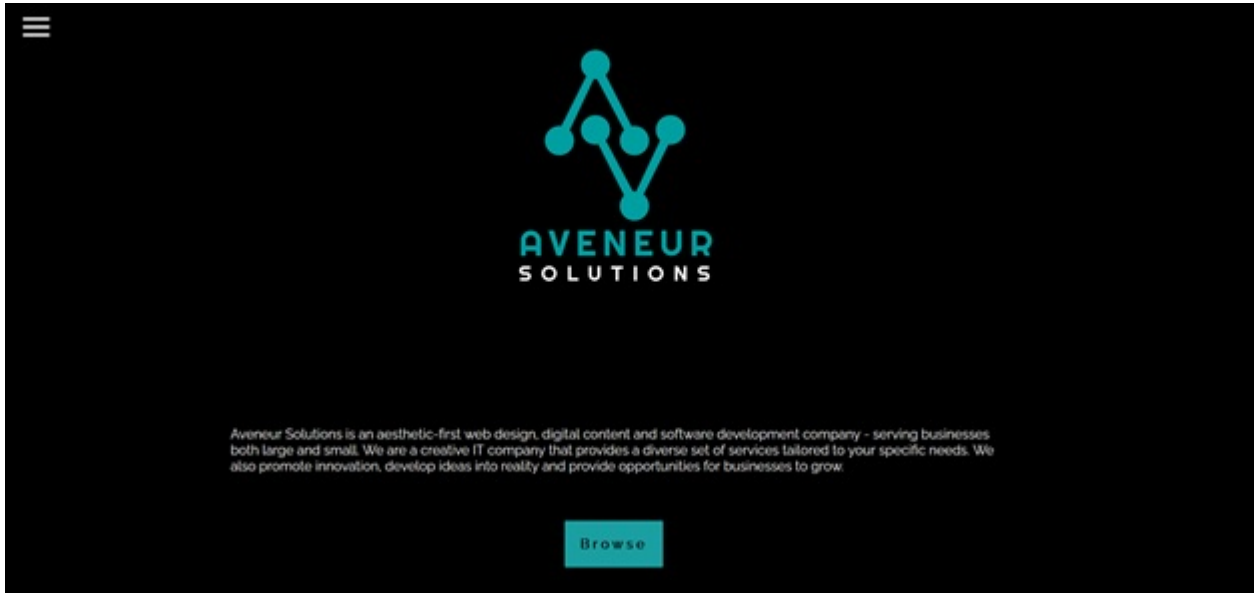


Figure 4.5: Senoark Concerns

Like Aveneur, the website needed the same backend functionalities, therefore we added an admin panel which looks similar, and functions the same ways as Aveneur's does.



Figure 4.6: Senoark Admin Panel

The **Functional Requirements** for the Senoark & Aveneur Websites are very similar to each other as they were made in the same timeframe and in the same way. The common requirements are:

- View Clients – Users should be able to see a slideshow of their previous and current clients.
- View Services –Users should be able to see the services provided by the companies.
- Location – Users should be able to view the location of the law firm in a google maps preview from within the Contact Us Page.
- View Blogs – Users should be able to view Blogs posted by the Senoark and Aveneur Team.
- Apply for Job – Users should be able to fill up a form and apply for a job opening.
- Manipulate Blogs – Admins should be able to add, edit and delete blogs.
- Post Job Openings – Admins should be able to post new job openings.
- Manipulate services – Admins should be able to add, edit and delete services.

Besides this, for Senoark, users should be able to see their sister concerns and sign up for startup incubation. Admins can manipulate the concerns and view the submitted forms for startup incubation. For Aveneur, users should be able to see what tech stacks we use and admins should be able to manipulate them.

The **Time Distribution** for both Senoark and Aveneur are shown below:

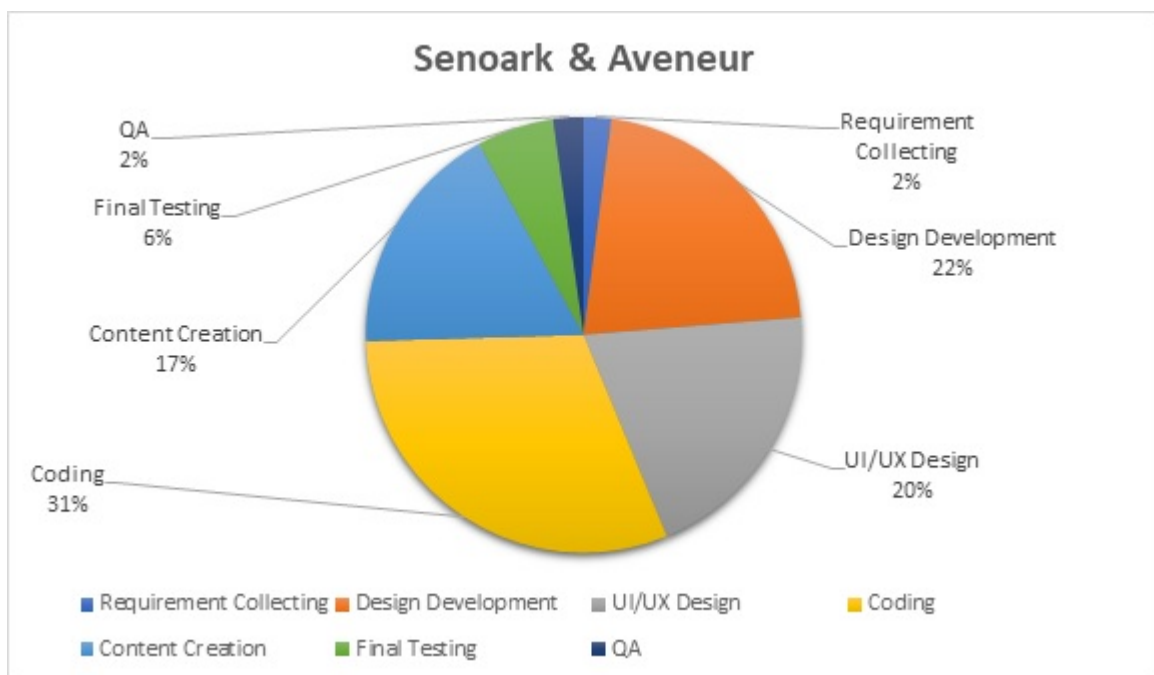


Figure 4.7: Senoark & Time Distribution

The Design Development, the UI/UX Design, and the content Creation were emphasized in this project, because we wanted our own sites to have a very modern look, that will attract new clients. Since both the sites were made around the same time and they are very similar in design and functionality, both of them have the same time distribution chart.

4.1.3 Your Lawyer App for LegalX

Your Lawyer App helps people find lawyers in their district according to their areas of law. The customers can also get free legal advice by starting a chat within the app with a Legal professional from LegalX themselves. Customers will be able to find lawyers close to them and will be able to filter the results according to areas of law and lawyer rank(band). They also have the option of adding a particular lawyer to their list of favorites. They can also have contracts drafted for them as it is a service provided by LegalX. They simply just need to fill out the proper form within the app and wait for its processing. Users also have the option to read the blogs posted by the LegalX team about legal news. They can also have a quick peek at the daily law tip the legal team provides and also look at the traffic laws whenever they need to. Both web application and mobile application are being made. For the forms in both forms, I have used the built-in validator function that

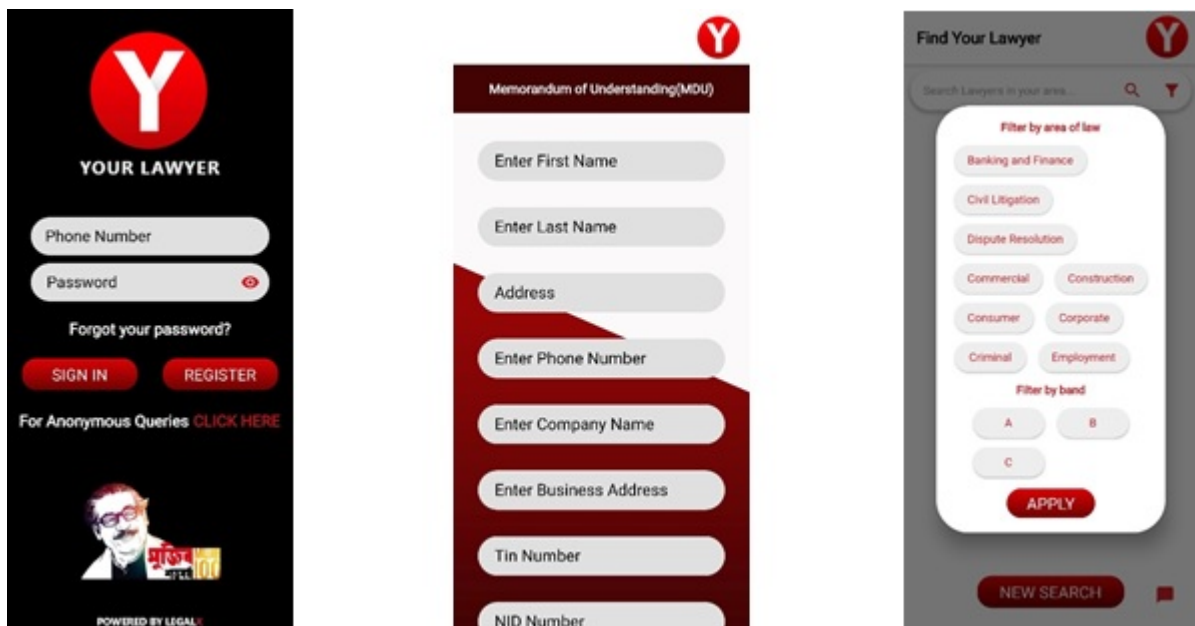


Figure 4.8: Your Lawyer App Screens

comes with the text field widget in flutter. In the login screen, when the user taps on the register button they're taken to the registration page. If the user taps on the "CLICK HERE" they're taken to the anonymous queries section where they can ask queries to the LegalX team anonymously. When the user taps sign in, an API call is made to the backend to verify the user's login credentials. If the credentials do not match, the back end sends an error response with the message "Login Credentials are invalid". When this happens, a toast appears at the bottom of the screen displaying that message. If the credentials are correct, the backend will send back a token as response. This token is important as it will be sent in the header of the

other API calls while using the app. This is what gives the permission to post or get information in the app.

For the forms, all of the forms have the same layout. The has to fill up the form and add his image at the bottom for identification and when he/she taps submit, an API call is made to the backend that takes in the form information as well as the form name, so that the form is stored in under the correct table in the database.

For the find lawyer screen, an API call is made to the backend to get the list of lawyers. Whenever the user types anything in the search box, another API call is made after the user stops typing for 500 milliseconds and lawyers matching the search criteria will appear below the search bar. When the user applies any filter, they are taken into consideration when calling the API again, therefore the results will change.

The admin panel has all the functionalities such as responding to client chats, look at drafting requests, update drafting tracker, manipulate lawyer info, manipulate traffic laws and edit the daily law tip. Below is a sample image of how the chats would look like from the admin panel perspective:

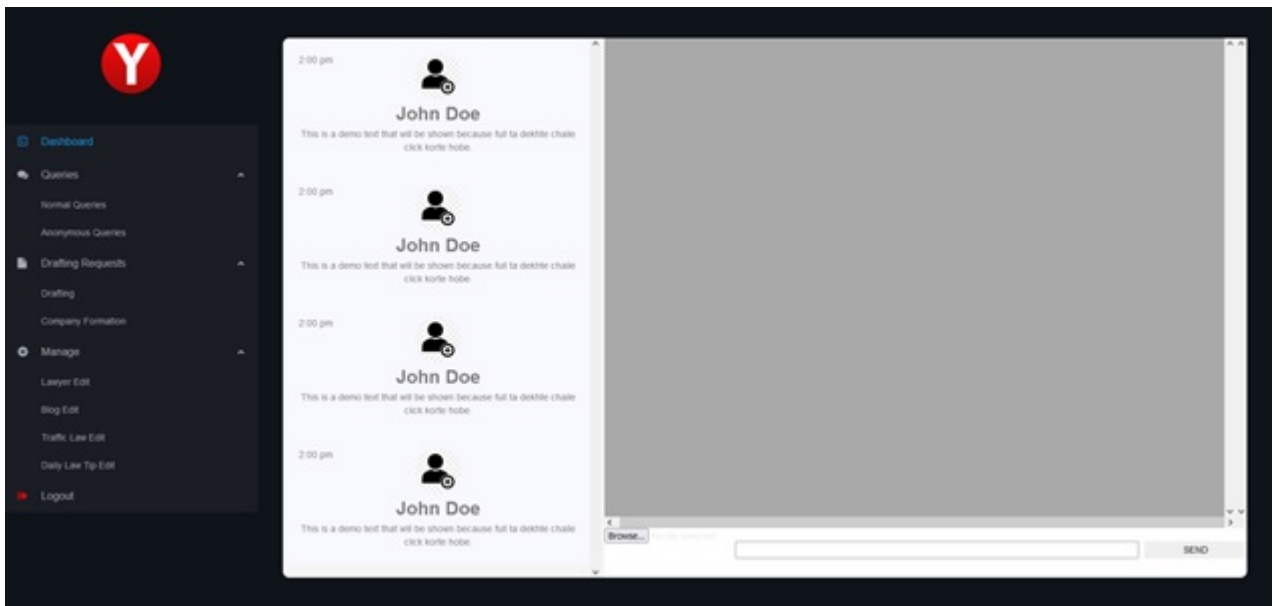


Figure 4.9: Your Lawyer Admin Panel

Since the app is still under development, there may be changes made to the final version of the app and the admin panel. The app was developed in flutter and the admin panel was made in react native. The backend was made with ASP .Net framework. Recently the decision to switch over to react native was made, therefore the existing app will be scrapped and a new app with the same layout will be made.

The **Functional Requirements** for Your Lawyer app is listed below:

- User Login and Registration – Users should be able to login and register a new account from both the customer end and the admin end.
- Password Reset – Users should be able to reset their password from both the customer and the admin end.
- Search Lawyer – Users should be able to search for lawyers in a specific district
- Filter Lawyers – Users should be able to filter the lawyers according to their areas of law and their bands (ranks).
- View Lawyer Details – Users should be able to view a lawyer’s information including his chamber address and contact information.
- Add to Favorites – Users should be able to add a specific lawyer to their list of favorites and be able to view favorites at any time.
- Get Official Contracts – Users should be able to fill up forms with information according to the contracts they want drafted. It should also be applicable for company formation.
- Access to Videos Regarding Rights – Users should have access to a section in the app with videos regarding public rights.
- Legal Advice – Users should be able to start a chat with a legal professional from the admin end to ask about legal advice.
- Contract Tracking – Users should be able to see the progress of their contract they have asked to be drafted.
- Lawyer Information Manipulation – Admins should be able to add, delete and edit a lawyer’s information.
- Manipulate User Data – Users should be able to edit their own data. Admins should be able to manipulate every user’s information when needed.
- View Contract Information – Admins should be able to view the information that a user has entered for his/her contract to be drafted.
- Update Contract Tracking – Admins should be able to update the progress tracker for a specific contract that has to be drafted.
- View User Chat – Admins should be able to use the chat history for all individual users.
- View Daily Law Tip – Users should be able to view a daily law tip and the home screen.
- Edit Daily Law Tip – Admins should be able to edit the daily law tip.
- View Traffic Laws – Users should be able to view traffic laws at glance for the hamburger menu.

- Manipulate Traffic laws – Admins should be able to add, delete and edit traffic laws from the admin panel.
- View blogs – Users should be able to view the blogs posted by the LegalX team.
- Manipulate Blogs – Admins should be able to add, delete, and edit blogs.

The Non-functional requirements for the Your Lawyer app are mostly similar to the main project except for the compatibility. Since it is a mobile application, and also a web application, the mobile application is only compatible for IOS and Android Operating Systems.

The **Time Distribution** for both Senoark and Aveneur are shown below:

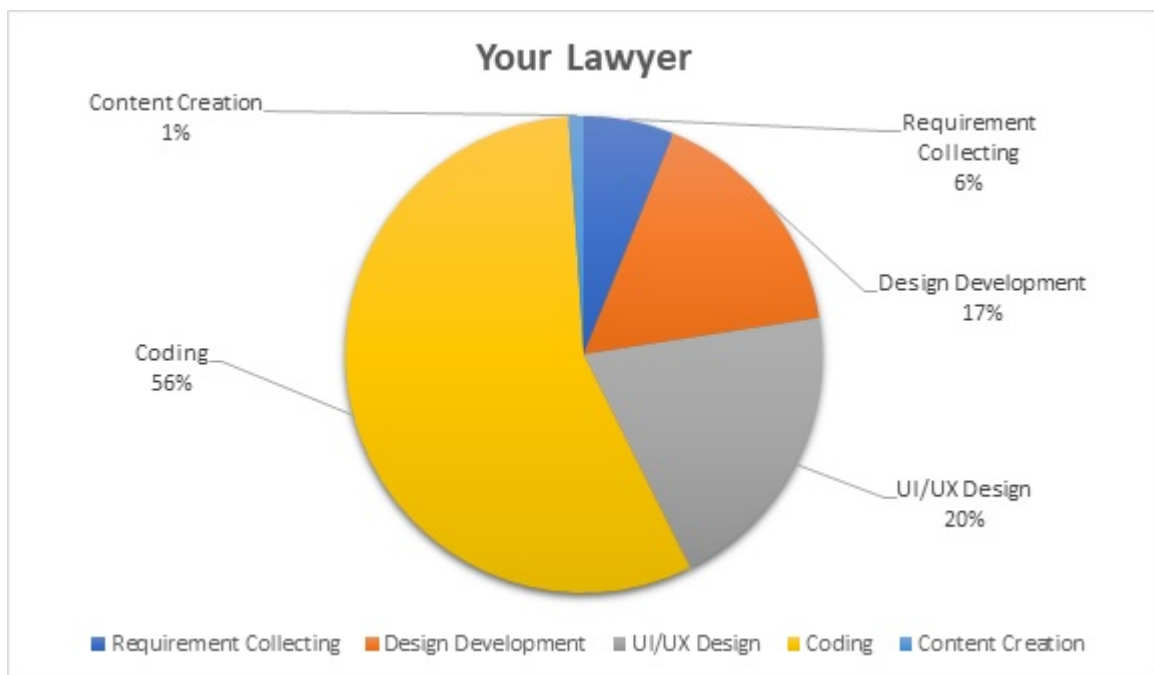


Figure 4.10: Your Lawyer App Time Distribution

For the Your Lawyer App, the content creation is as low as 1% because there was not much content to make as the app mostly provides services that will be handled by a legal team. The UI/UX took more time as we had to think the most about ease of use. The app design should follow a design similar to well-known apps in the app store so that people will instinctively know where to locate certain features and settings. As for the missing QA and Final Testing, the app is still in development, therefore the time distributions for them could not be provided.

4.1.4 Portfolio Website for Shawn Novel & Associates

Shawn Novel and Associates is another law firm which is looking to upgrade their existing website made by WordPress, into a fully customizable React website. They saw that whenever they wanted to change out the design of their site, they would encounter a lot of problems in WordPress. They also could not get a theme that suited their website, therefore they wanted to rebuild the whole website from scratch. In the Website, users can see the details of the services they provide, the blogs they post, and see their list of previous clients. They can also see the details of the lawyers in the firm.

The site is still under development but below is a sample picture of the home page:

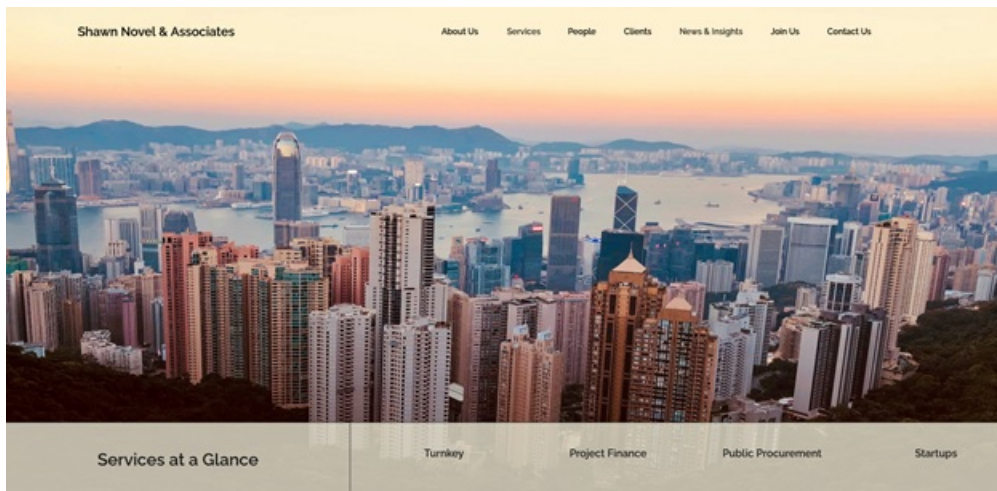


Figure 4.11: Shawn Novel & Associates Homepage

For this website I have designed the About Us page, the service page and the News & Insights Details page. All of them have the same layout so I only showing one image:

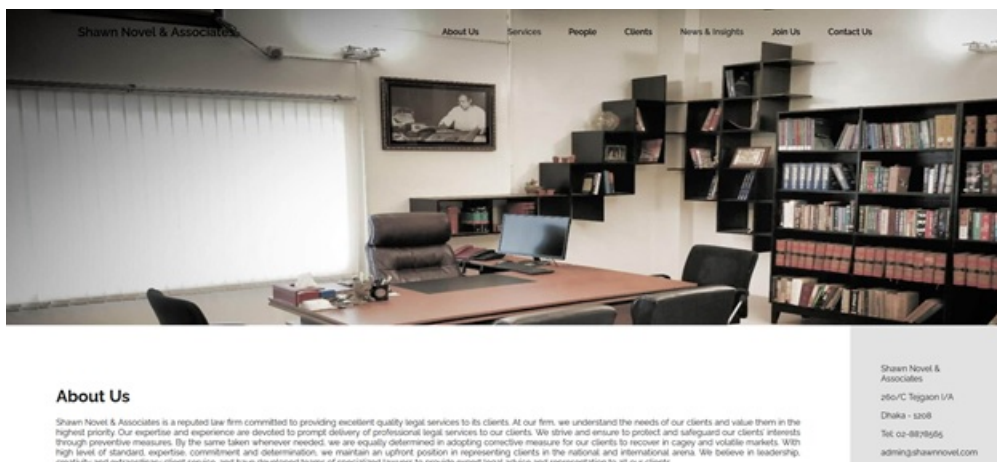


Figure 4.12: Shawn Novel & Associates About Us

The image position is fixed so when the user scrolls, the contents of the page go up and cover the image. The About Us page is static, but the service details page and the news details page call APIs on load. They send requests to the backend server and get back the title, image URL, and the description as response.

I have also made the admin panel using Django and Django Rest Framework. Since they only require the option to add services, news and clients, we did not make a full admin panel for them and used the default Django admin panel. This saved a lot of time because we did not have to design and code the admin panel. Below is a sample image of the admin panel:

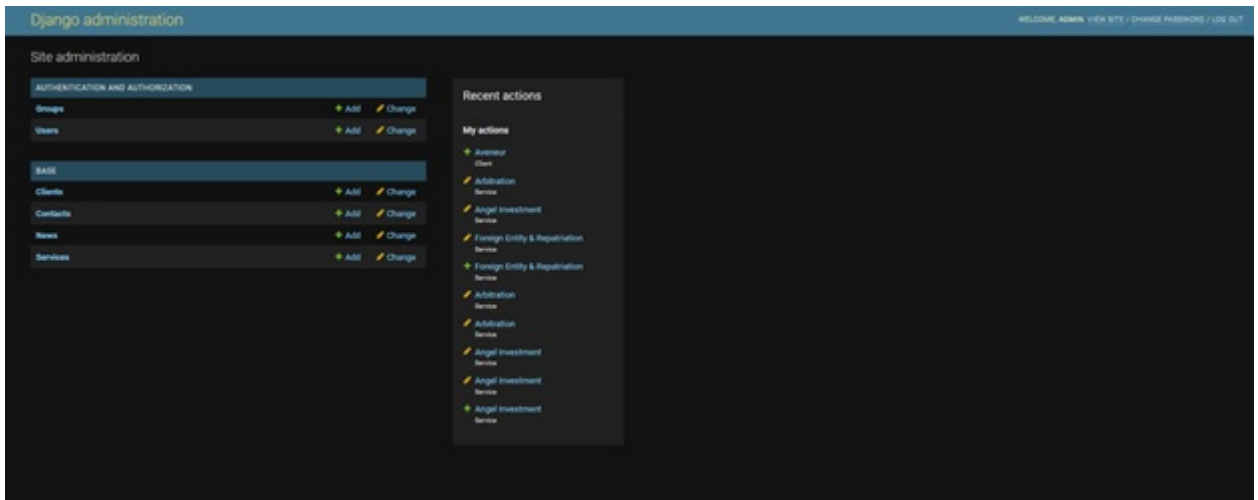


Figure 4.13: Shawn Novel & Associates Admin Panel

I had to make models for the Clients, News and Services, which would become tables in the database. The Contact Us is temporary as it might get removed later if we decide to not use EmailJS for the email service. With the models, I had to create model serializers to specify which fields to send as response. For the API I had to specify whether to send all the data from a table or 1 instance, for the detail view. For the news and insights, I had to paginate (send only a few objects per page) the data as there would be a lot of news blogs, and it would be very inefficient to load all of them in one page.

The Functional Requirements for Shawn Novel and Associates is listed below:

- View Services – Users should be able to view the services and their details provided by the law firm.
- View Clients – Users should be able to see the lists of clients of the law firm represented by their logos.
- View News and Insights – Users should be able to view the news and insights provided by the law firm.
- Contact – Users should be able to directly send a mail to the contact provided by the law firm from within the website.

- Admin Login and Registration – Admins should be able to login or register to access the admin panel.
- Manipulate Services – Admins should be able to add, edit and delete the services and their details.
- Manipulate Clients – Admins should be able to add, edit and delete client logos.
- Manipulate News – Admins should be able to add, edit, and delete News and Insights.
- News Ticker – Users should be able to view carousel of news about Law right in the home page.
- Location – Users should be able to view the location of the law firm in a google maps preview from within the Contact Us Page.
- View Lawyers – Users should be able to view the details of the lawyers working at the law firm.

The **Time Distribution** for Shawn Novel & Associates is shown below:

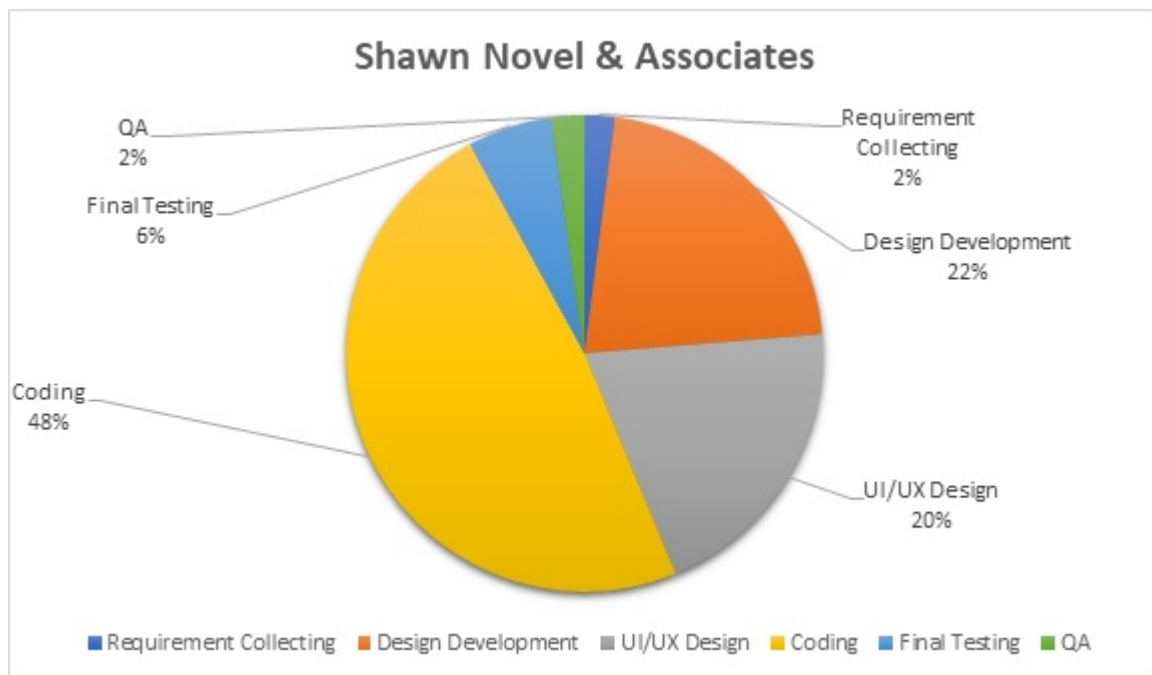


Figure 4.14: Shawn Novel & Associates Time Distribution

The time distribution for this website is similar to that of Senoark and Aveneur’s, except there was no need for content creation as all the content was given to us by the law firm themselves. Most of them were taken from the previous website.

4.2 Main Project I Have Worked On - Homeland

4.2.1 Project Description

Homeland wants to make a platform where people can book flats within their own residential society. Through their website, people can go through each flat, look at their interior designs and see the location of the flat in Homeland's society map. If they see a flat, they like within their price range they can book the flat for 10% of the flat's price as down payment. The booked flat will be converted to an allotted flat once the payment has been made. If they change their minds, they have the option to cancel booking, or get a discount on their next booking if they transfer their booked flat to someone else. Users simply have to register with their phone number and pay through the payment gateway integrated into the website. Below is a sample images of the website:

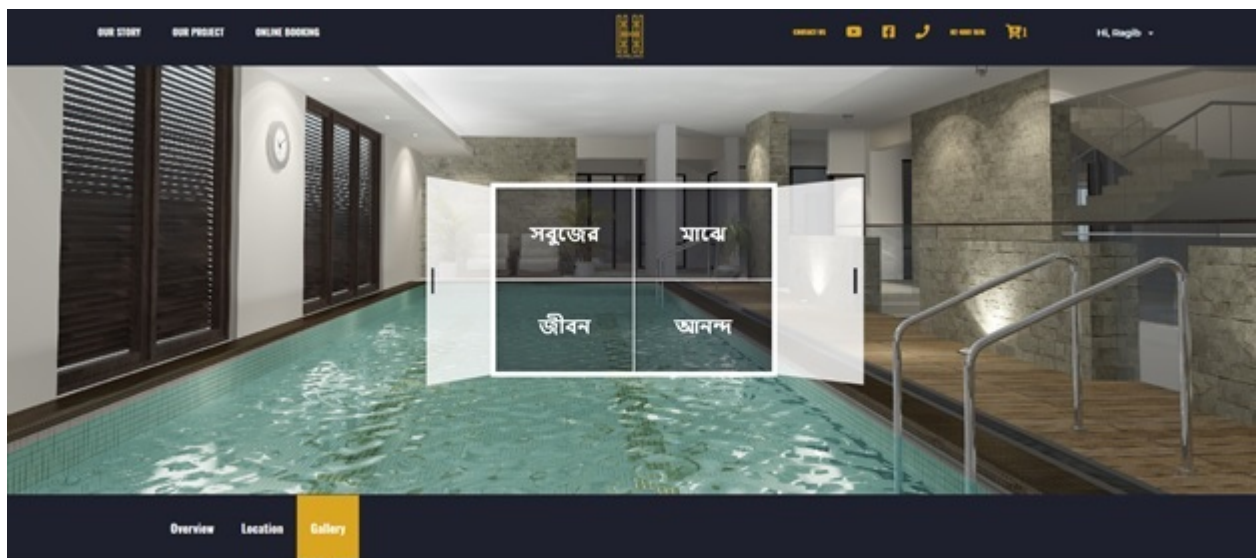


Figure 4.15: Homeland Homepage

The admin panel, like the client-side website, was made using React for front-end, and ASP .Net Framework for the backend. It has all necessary tools to manage dynamic content like the gallery, the flats for sale, customer management, booking management, allotment management, and payment management. It also shows an over view of the total number of flats available, number of booked flats, number of flats available for booking, number of flats transferred and total number of users in the site.

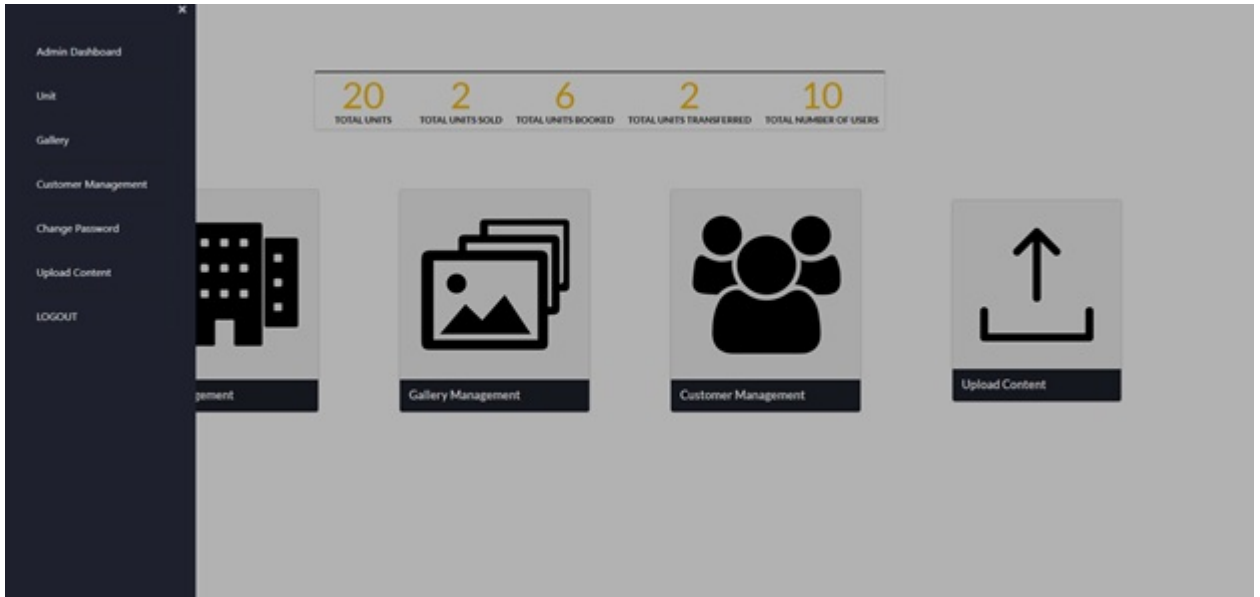


Figure 4.16: Homeland Admin Panel

Both the website and the admin panel are fully completed. The final QA testing has been initiated. It is supposed to be handed over to the clients within the first week of August.

4.2.2 Project Breakdown Structure

A project's breakdown structure is represented by a visual guideline, and a hierarchical overview of the project. We can create tasks by breaking down the deliverables and these can again be broken down into subtasks as many times as needed to reach the goal of the project. Most of our projects follow a similar structure which is portrayed below:

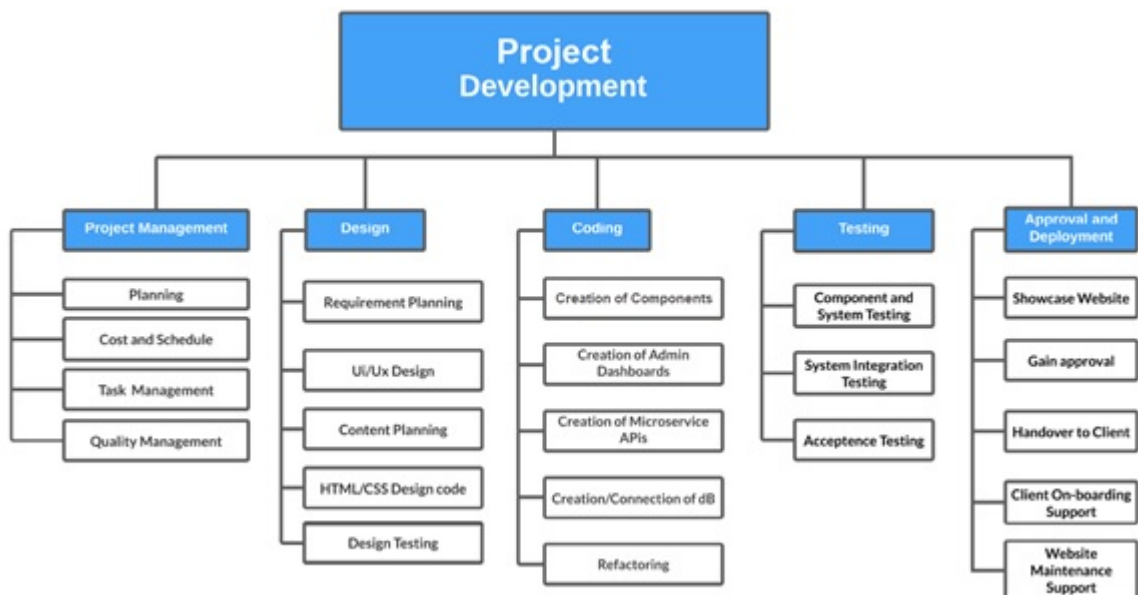


Figure 4.17: Project Breakdown Structure

In case of mobile development, we cannot generate HTML/CSS design code since the project we have worked on is done on flutter, the designing has to be done form looking at the UI/UX design.

4.2.3 Job Responsibilities

For the Homeland Site, I have made 2 pages. The Profile Page, and the Flats Page. The profile page does not appear in the navbar above if the user is not logged in.

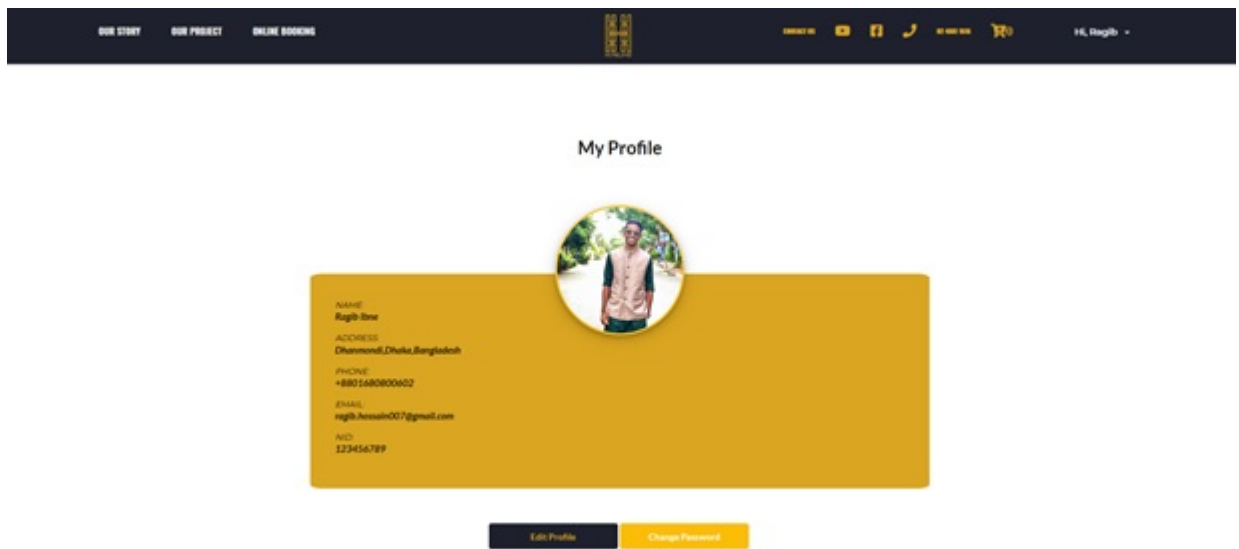


Figure 4.18: Homeland Profile Page

When the user logs in, a token is sent back from the back end for authentication purposes and is stored in the local storage of the host. This token is required to send API calls by only authenticated users. When the page loads, an API call is made to the backend with the token as a parameter. The backend server deconstructs the token to identify the user and sends a response with data like name, address, email, NID and user photo URL to the frontend. These are stored in models, and is stored in the MobX user store. This is done to make the component's state easier to handle. The requirement of models come here as the website was developed using TypeScript rather than JavaScript. This is done so that the testing phase becomes easier, as with TypeScript, errors are identified much earlier. When the User presses the 'Edit Profile' button, A form fades in from the bottom and replaces the profile info component. There the user can edit the credentials he/she put down except his/her NID. The form validation for all the forms in the site was done using React Hook Form. After the user presses the save button, an API call is made to the backend to replace the information stored in the database as a PUT request. When the user presses Change Password, a form pops in a similar way to the previous form, which tells the user to enter his/her old password, new password and confirmation for the new password. A similar API call is made as the previous one, but this time, before the call is made, a validation check is performed to check whether the new password and confirmed password match. If not, the warning is shown below the textboxes. When the API call is made, another check is performed by the backend to see if the new password meets the criteria for being secure. In both cases, the

backend sends a response to the frontend with either a success or error message. Depending on that, the message is shown as a toast at the bottom right. This was done using react-toastify.

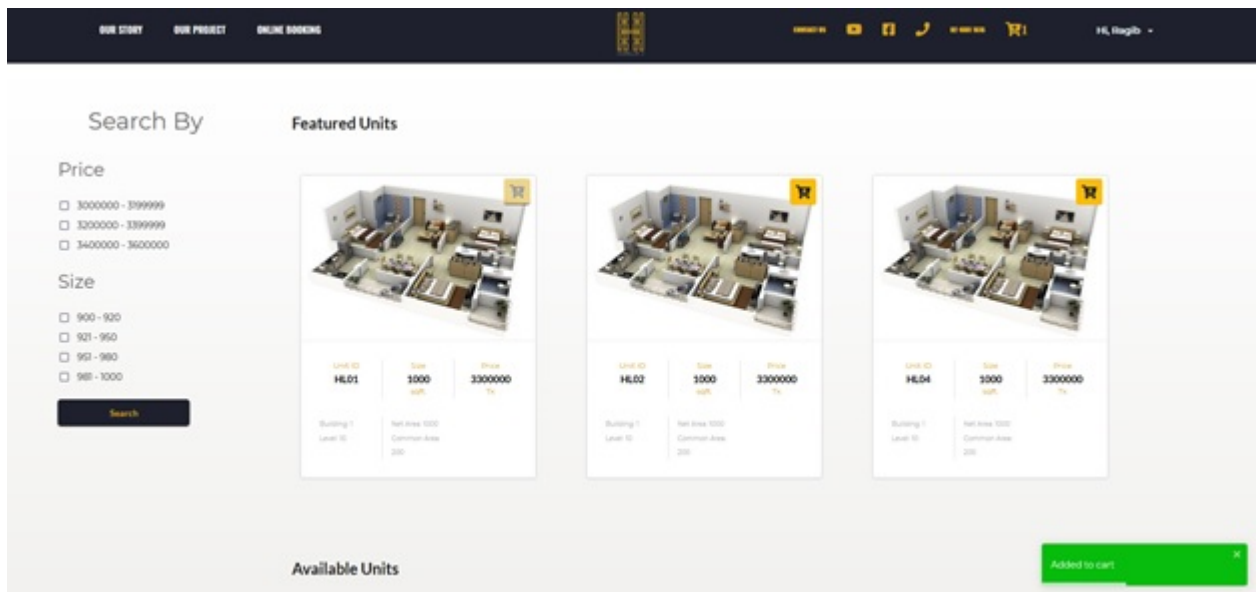


Figure 4.19: Homeland Flats Page

For the show flats page, on page load, two API calls are made. The first one is to get the featured units list, and the other one is to get the list of all available units. When the user clicks on any one of the units, they are navigated to the unit details page. The user can click add to cart on the top right of any unit from this page, but to do so, he/she must be logged in. The options on the right will apply filters and perform the search using those parameters.

4.2.4 Functional Requirements

The functional requirements for Homeland are listed below:

- User Login and Registration – Users should be able to login and register a new account from both the customer end and the admin end.
- Book Flats – Users should be able to book flats on the customer end. The booked flats will be added to their cart.
- Manipulate Booked Flats – Admins should be able to make manipulations like add, edit or delete the booked flats from all customers.
- User Payment – Users should be able to pay for booked flats by checking out their cart. Booked flats that are paid for should be turned into allotted flats.
- Allotted Flat Manipulation – Admins should be able to manipulate the allotted flats.
- Transfer Flats – Users should be able to transfer allotted flats to another person (each flat only once), and get discount on his next booked flat.

- Transferred Flat Manipulation – Admins should be able to manipulate transferred flats.
- Filter Flats – Users should be able to filter flats according to price range.
- View Flat Details – Users should be able to view a specific flat’s details.
- View User Info – Admins should be able to view a customer’s details.
- Edit User Data – A customer should be able to edit his personal information entered when registering. Admins should be able to edit all user’s data except a user’s password and registered phone number.
- Reset Password – A user should be able to reset their password on both the customer end and the admin panel.

4.2.5 Non-Functional Requirements

The non-functional requirements for Homeland are given below:

- Performance – The website should load images, CSS and JavaScript Files fairly quickly. It should also be able to make requests to APIs quickly as well. In this day and age if a website does not load in 2-3 seconds, people will lose interest in the site and this will lead to decline in customer quantity. This is the main reason WordPress was not preferred for this website.
- Compatibility – The website should be compatible to all the major operating systems being used today. This includes Windows, Mac, Linux, Android and IOS.
- Responsiveness – The website should be made responsive for most resolutions i.e., standard 1080p and 720p for pcs, custom resolutions for all phones and tablets.
- Security – Since users are able to make payments from the website, it should have very secure connections. The APIs are to be called from links with SSL certificates. It should also send an OTP to users every time they login or register.
- User Friendly UI – The UI should be easy to navigate. Too many clicks result in bad user experience, therefore the UI should be designed in a way which minimizes the number of clicks a user has to make to perform a specific task.

4.2.6 Time Distribution

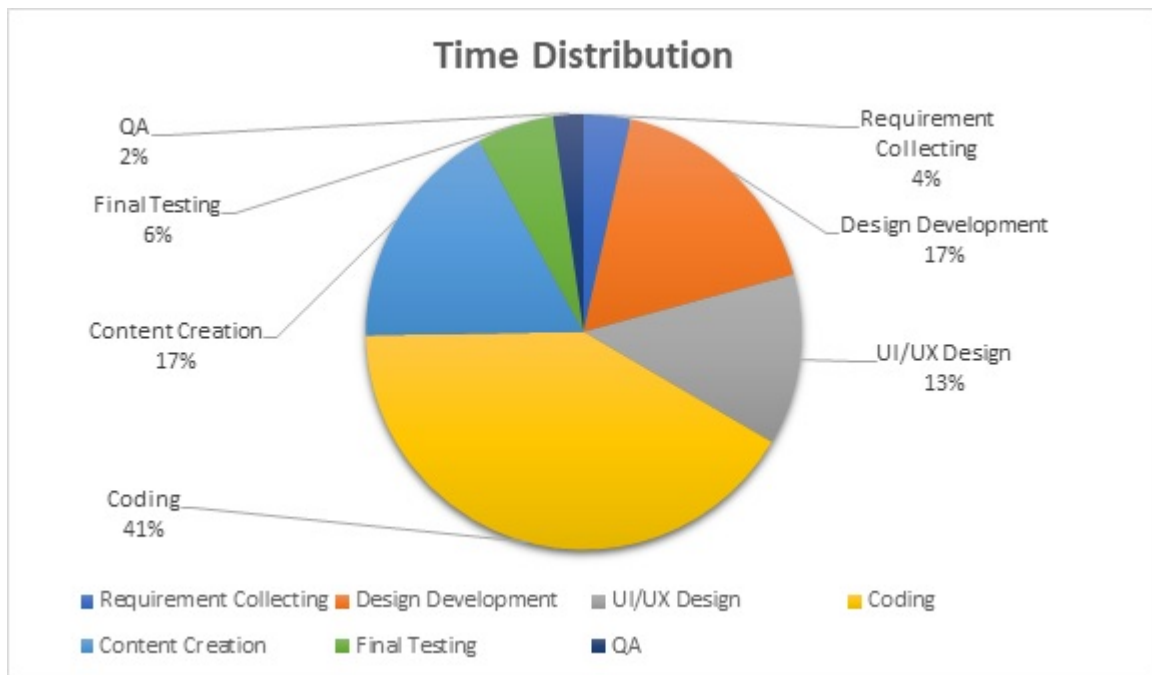


Figure 4.20: Homeland Time Distribution

In this project, apart from Coding, Content Creation took the most amount of time as Homeland had a lot of ideas about the content and we gave them constant feedback about the contents created. The logo animation especially took more than a few tries to perfect.

4.3 Tools used

We have used a number of tools in the project that will help with different aspects. The most common tools used are listed below with a brief description of each tool:



- We have used Adobe XD to design UI/UX of the web pages first. Adobe XD allows us to create layouts for both the Web view and mobile view of the project, so that we would not have to worry about website responsiveness later on. It is very intuitive and easy to make the layouts and connect them to make a UX as well.



- We have used Visual Studio Code as our editor. Even though Visual Studio itself has an IDE for ASP .Net core projects, we chose VS Code instead because it is much more lightweight. Visual Studio Code can also be used for any other language. It has an extensions panel where there are various extensions available which help in a lot of different aspects of coding. Starting from fixing

code indentations, code auto completion for specific languages or frameworks to visual upgrades which help identifying closing brackets; it has everything. It also has its own integrated terminal, so it can also work as an IDE.



- For version control, we have used GitHub as everyone in the development team is familiar with it. Through it we can create repositories and upload our code to the repository. Each change we make to the code, we “commit” them and “push” them to the repository. GitHub allows us to roll back to a previous version of the code/ a previous “commit”, if we see any problems in the new one. A lot of people can work on the same project, so to avoid confusion and code mismatch, different branches can be created for different modules of the project and can later be merged altogether in the main branch.



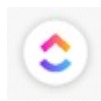
- For deployment we followed the Continuous Integration Continuous Deployment (CICD) Practice and have used GitHub Actions to continually deploy each of our project increments. Every time we build and project and push the newest version to the main branch, an automatic script runs where the Website can be deployed into the hosting right from GitHub.



- To test APIs, we used Postman. It allows us to send API requests to the backend server directly within the software itself. It has a lot of useful features like attaching an authentication token to the request and setting variables that might be used for a lot of different APIs. With it we can also document APIs and save them as a collection so that the collaborating team can have easy access to all the APIs they need for their project.



- To monitor changes in state and variables in React, we have used the chrome extension called React Developer tools. React Developer Tools can identify when a page is using react and it can display the structure of the components used in that page. Within these components are information such as their state and props. These help us to see what values are changing when interacting with the site in real time.



- To keep track of tasks we have used ClickUp. Through it, tasks can be created with deadlines for specific tasks, and different people can be assigned to different tasks. As we have used discord as our primary method of communication,

we have connected the ClickUp bot to discord, so that when a task is created or completed, a notification is sent to the discord channel.



- For state management, we use MobX. Many times, it is difficult to keep track of state, especially when the state of a component is being changed in another component. This is where MobX helps. It is a central store in which states can be stored and mutated. It also helps keeping the code clean.



- For site navigation, we use React Router. It is a collection of navigational components with a lot of useful features which make navigation to other components very easy.



- To make API calls and consume them, we used axios. Axios is a Promise based HTTP client for web browsers and Node JS. It much easier to use and is much more flexible than JavaScript's default Fetch API.



- To send automated emails, we have used EmailJs. We used Email Js in most of the 'Contact Us' Pages in our portfolio websites. This will send what a user has written in the provided fields and will send that to the desired email configured by the website developers.



- Even though we have coded in VS code, we needed to emulate our apps in an Emulator. For Android Emulation we have used the emulator that comes with Android Studio. For IOS emulation, we have used the emulator that comes with XCode. This software is only compatible in MacOS.

Chapter 5

Project Teams

During my time in Aveneur, I had learned how to work with a team while working on production level projects. Being a part of a team, I could improve my skills drastically faster than I used to, through team building exercises. We could work on projects with a good pace with proper team assembly and, we always kept our morale high by celebrating and participating in company events.

5.1 Team Assembly

For every project, there would always be 2 separate teams. One is for the front-end development, and the other for the backend development. For most of the projects I was in the front-end side alongside Mr. Zulker Nien and Mr. Faisal Bin Alamgir. For the backend, it mostly Mr. Ragib Ibne Hossain, Mr. Ashikur Rahman, and Mr. Zunaid Amin Enan. The testing would be done separately within these teams. Our Project Manager Mrs. Faria Tuz Zohora, is the person who assigns teams based on their skill on technologies they have learned. The teams I have mentioned were not strictly confined to front end or backend. When the one team needs help, the other team is always there to support, as most developers who worked in Aveneur Solutions were Full Stack Developers. Also, everyone could not participate on every project. For example, for the Shawn Novel & Associates Website, Only Mr. Zulker and myself were assigned so that our focus would not deviate towards other projects and we could finish the project rather quickly.

5.2 Team Building Exercises

Knowing team members and how to work with them is very important in an office environment as most of the time, one person alone will not be assigned to a single project. If employees work alone without informing their teammates about the task they are working on, there might be overlap in the tasks, or worse, code might get merged into something which is completely unusable, and as a result, it has to be programmed from scratch again. To avoid issues like this, one should get to know their teammates and be in constant communication while working. They should try to figure out their style of coding, which will help in deciphering their code faster. They should clearly separate tasks so that there is no chance of overlaps and they should always try to follow the same pattern of coding which will later help in de-

bugging code.

At Aveneur, we perform various team building exercises. One of them is holding informal competition against each other to see who can finish their work the fastest without any errors. This encourages employees to be more productive and recognize each other's strengths and weaknesses. This way, when working on projects, one team member can cover for another team member's weaknesses.

5.3 Team Events & Celebration

As a team we work hard and occasionally celebrate on a job well done, increase morale in the workplace. More often than not, the whole Aveneur team would go out to eat during lunch at the closest food place, and occasionally after working over-time, we would go to a restaurant that we all agreed upon. We have even planned to go to Cox's Bazaar when the COVID situation is more in control. As a part of Senoark, the Aveneur team has participated in the Senoark Football Cup 2021 which took place in 8th of May. All teams were assembled through a lottery system. I was a part of Team Thunderwolves, where I thankfully saw a familiar face from Aveneur – Mr. Faisal. We played against two teams, but we unfortunately lost both of the matches as none of our players had played football since the last 3 years, but the loss did not stop us from enjoying the event. We had a combined photoshoot of the Senoark and Aveneur teams after all the matches ended.



Figure 5.1: Senoark Football Cup

After the event all of us went out and got pizza, and spent a few more hours together. Another event is coming up in October which will be the Senoark Cricket Cup 2021. All of us hope to be there as well.

Chapter 6

Professional and Technical Growth

The time I have spent in Aveneur Solutions has resulted in growth from both career and skill perspectives. There are a lot of factors which overlap in the two categories, but in Professional Growth, what will be discussed about most is how my time in the industry helped me improve my attitude towards professional work. In Technical Growth, I will mostly talk about my improvements from a programmer's point of view.

6.1 Professional Growth

There are a number of ways that I can say I have grown professionally, but for that to occur, I had to deal with many challenges I've faced along the way. The professional growth that I have achieved can be categorized according to those challenges I have faced.

6.1.1 Adaptability To The Office Environment

The whole office scene was very new for me so I had to learn how to adapt to my surroundings. This included many things such as matching my schedule to the office time, adjusting my attitude towards the office employees, having lunch and spending break hours with them, and learning how to work with each and every one of them. I eventually had adapted to the office environment quite well, and would have done so sooner if the pandemic situation had improved.

6.1.2 Adaptability To Remote Work Environment

Due to the pandemic, there were a times that we had to work from home because of the city-wide lockdown. It was actually harder to work from home as I had difficulty trying to not get distracted by everything. While working, a lot of times I would have urges to play games, lie down on my bed or even casually browse the internet as my work was not being monitored by any software like Microsoft Teams. Every day, we had to be present in Discord voice chat by 10 am and say that we were present. We generally had a meeting about what work was being assigned to whom and then just continue working at our own pace. However, this did not mean that we could slack off as our work progress was being monitored by our GitHub pushes and pull requests. We also were using ClickUp to keep track of every module being worked

on and also employees' progress. We were to update ClickUp any time we finished working on a particular module. We had integrated both ClickUp and GitHub bots to discord so anytime there were pushes or pull requests to any GitHub branches, and any task being created or completed in ClickUp, that would automatically be logged into Aveneur's Discord server's text-channel. I always tried my utmost to not give in to the laziness and eventually got accustomed to working from home.

6.1.3 Punctuality

Due to most of my classes beginning after 2 pm, I had grown accustomed to waking up late, usually after 11 or 12 pm, but as soon as I started going to the office, I had to wake up at 8 am. It was very hard at first, so much so I had gone more than 24 hours without sleep just to fix my sleep cycle. After a few days of sleepless nights, I was always waking up at 8 am and was in office by 9:30 am.

6.1.4 Maintaining Deadlines

Before joining a company, any projects I did, I used to do at my own leisurely pace, but at Aveneur, I had to work on production level projects therefore I had to maintain strict deadlines. There would be many consequences if I missed a deadline, for example, another co-worker's work might be halted because his module depended on mine, or worse, we might not be able to submit the project within the time promised to the client. At Aveneur, customer satisfaction was always given a priority, therefore we could not ever let that happen. Therefore, from day 1 in the company, I had to learn how to work for straight hours without taking the very long breaks which I had gotten used to before. After a week, I had gotten used to the pressure of deadline so I did not have any problems any more.

6.1.5 Communication Skills

Effectively communicating with others was the biggest challenge I had faced in the workplace, as I was very introverted and had a hard time normally communicating with people whom I did not know. There were times where I would freeze up and keep stuttering when I got too anxious or nervous, but luckily some of my coworkers were very friendly and they reached out to me and told me to take it one step at a time. I took their advice and I started to get to know each of my co-workers better, so much so that the nervousness would fade away and finally I was able to communicate normally with them.

There are other etiquettes that I had to learn which can be categorized under communication skills as well, for example: learning how to write formal emails. Normally I would always write emails whenever I needed something, but the people in office would always be busy therefore the time between sending an email and getting a reply would always be long, therefore I had to take my time writing the email as I would try to fit in whatever I wanted to know and ask questions in such a way that the email receiver could reply to them directly to the point. There was also the habit of writing a report of my work progress. At the end of every week, we had to write a report detailing what I did for that week and send it to the product manager so that

she could keep track of my performance. It was tedious, but necessary, therefore I had to make a habit out of it.

The most important growth I have achieved was being able to talk to clients. Given my lack of people skills, this was something tremendous. Clients, especially if they are non-technical, would always ask for something more, but the part that was concerning was that they would ask for them in a ridiculously small timespan. I had to figure out how to keep them happy without telling them that what they were asking for was impossible. This took a lot of practice and I would often shadow my superiors to see how they deal with clients. In the end, I was finally good enough to deliver a product to a happy client.



Figure 6.1: Successful Handover

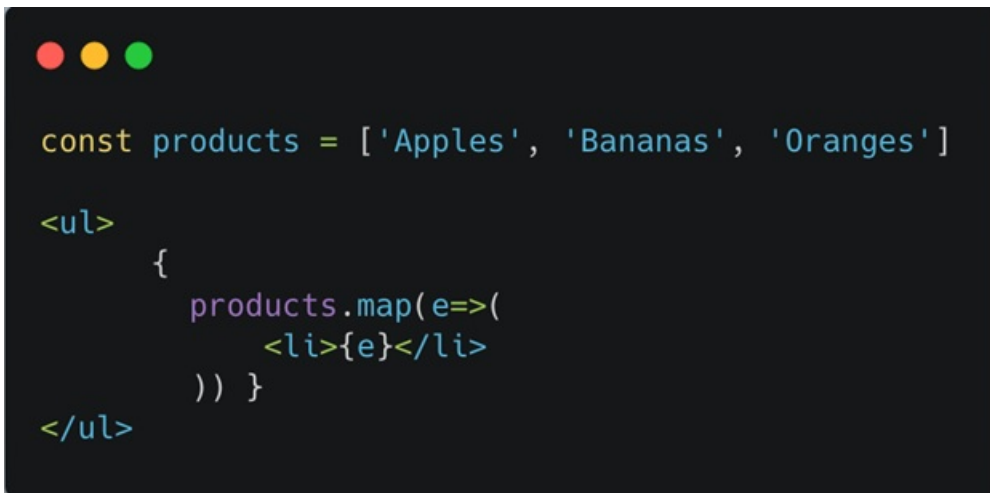
6.2 Technical Growth

I came to Aveneur seeking to improve my skills, and after a long journey I can happily say that I am a much better programmer than I was before joining the company. The technical growth that I have achieved can also be categorized, mainly into the

proficiencies of the language itself, the frameworks used and the different patterns used to approach a problem.

6.2.1 JavaScript Proficiency

Before I had joined Aveneur, I had already known how to make a webpage using basic JavaScript, HTML and CSS. I thought I was already proficient in it, but that wasn't the case when I found out that I still had a lot to learn. I had learned how to properly make use of the features added in ES5 (ECMAScript) and ES6. The main feature which had been of so much use was the `array.map()` loop. This takes in a function as a parameter and that function takes in an element as a parameter which signifies each individual element in the JavaScript Array list. With each element, the same operation can be performed on it and can be returned that way. For example, if there is an array of products which are to be listed in `li` tags in HTML, it would be as simple as writing:



```
const products = ['Apples', 'Bananas', 'Oranges']

<ul>
  {
    products.map(e=>(
      <li>{e}</li>
    )) }
</ul>
```

Figure 6.2: JavaScript `Array.map()`

There are also other features like `array.foreach()`, `array.filter()`, etc. but `array.map()` was the most useful one by far.

6.2.2 Framework Proficiency

I learned about programming frameworks whilst in Aveneur, and it opened up a vast new side of web development for me. Before I only knew that writing PHP and JavaScript monolithically was the only way to develop websites. After I came to know about frameworks everything just became so much easier and code became so much cleaner. While it did take me 1 month to learn ReactJs for front-end, I had yet to learn about the best practices for it. I had a lot of help from the company's developers who taught me to the best ways to use the component life-cycle methods, and later on, functional components. Just like how JavaScript is being updated every day, React is also being updated, therefore I came to know that class-based components were rarely used anymore since the birth of 'React Hooks'. Hooks like `useEffect()` could be used as component life-cycle methods. I also came to know about `PrivateRoutes` where a `Route` in the Web Application can only be accessed

only when the user is authenticated. This made the main component where all the routes are stored much cleaner, as all the logic was being handled in another component.

As for Flutter, most of the developers were also new in flutter that is why the decision was made that the stack would be shifted to React Native, since it was so similar to ReactJs.

For back-end, the backend developers had already taught me about how MVC patterns work and, giving me detailed explanations for the logic behind them. Although I did not get to work much with ASP .Net Framework, I did get to work with Django and Django Rest Framework, where everything was made even easier. Making APIs was not much of a hassle with Django because Django Rest Framework decreases most of the boiler plate code that used to be required for Django. Now it was just a matter of making a model for the database, making serializers for that model, making an API endpoint and specifying whether data should be Created, Read, Updated or Deleted through that endpoint. The serializers were there to format and validate the data in the models.

6.2.3 Different Approaches To A Problem

In programming there are a lot of different approaches to a particular problem, and while I knew that, I did not know of the many approaches one might take in developing a web application or mobile application. Throughout my time in Aveneur, I have learned a number of approaches to many problems. The one that stands out the most is I would say the Axios configuration file where all the API calls in the front-end were being handled. What I have seen in many of the tutorials is that developers keep writing everything in a function for a single API call. That would make things very tedious if there were a lot of API calls and there would be a lot of repeated code. What I learnt was to make four separate functions for four types of calls - GET, POST, PUT and DELETE. These four functions could handle any API call, respective to their type, with the endpoint passed down as a parameter, and return the response. This would make API calls much easier to read and would reduce a lot of the repeated code as it has been made so much more modular and re-usable. This is how it would look like:

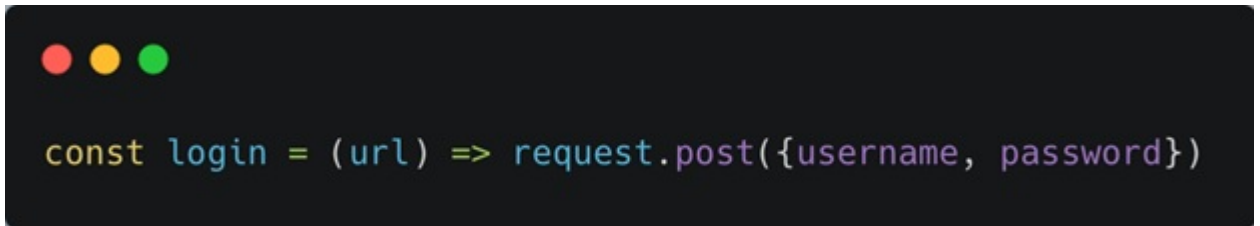


```
const responseBody = (response: AxiosResponse) => response.data;

const request = {
  get: (url: string) => axios.get(url).then(responseBody),
  post: (url: string, body: {}) => axios.post(url, body).then(responseBody),
  put: (url: string, body: {}) => axios.put(url, body).then(responseBody),
  del: (url: string) => axios.delete(url).then(responseBody),
};
```

Figure 6.3: Request Functions

From here, if an API call needs to be made, then it would look like:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in a light-colored font with syntax highlighting:

```
const login = (url) => request.post({username, password})
```

Figure 6.4: Login Function

This looks so much cleaner than:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in a light-colored font with syntax highlighting:

```
const login = () => axios.post(endpoint, {username,password}).then((response) => {  
  user = response.data;  
})
```

Figure 6.5: Messy Login Function

Chapter 7

Self-Assessment Suggested Improvements

In this section I have written about the my assessment of the internship program, and also some issues I observed from both Aveneur's side and BRAC's side. I have written about the improvements that can be made for those issues as well.

7.1 Self Assessment of Current Internship Program

While I was in the care of Aveneur, I had access of many different facilities. For starters, I could always have free lunch in the office and I had my own computer. For two months this was the only facility I had because I was under a probation period, so that they could see if I was able to keep up with the tasks that were handed to me. After the two months, I was paid 15,000 taka monthly for my work. They also used to pay for any ride fare that was related to work, like going to client meetings. As for leaves, I was granted 10 days of paid leave and 16 days of unpaid with valid reasoning every year. The most important facility to me was free access to udeemy courses, so I could gain knowledge on front end and backend whenever I wanted.

7.2 Suggested Improvements for future internship programs

Firstly, I noticed that the developers in Aveneur have to do a lot of menial work, like making a webpage and making basic APIs. For this, I would recommend Aveneur to reach out to BRAC and take in Interns periodically. They would have an extra set of hands to handle the more menial tasks, so that their own developers can spend more time in more important things like putting in more time in developing in-house projects. If they develop a template from which interns can learn from, they would not have to spend much time in teaching the interns either. The template should have a project that all interns are required to complete so that they can understand at least the basics of what they will be working with.

As for BRACU, I asked a lot of my colleagues and fellow classmates about whether they were doing Thesis, Project or Internship, I was astonished to find out that none of them knew that Internship was an option for CSE students. All of them thought that it was only an option for BBA students. This is why I think BRACU should make the students more aware of the Internship Program. I believe that Internship would be more beneficial for students who do not want to pursue higher education for some time and want to get into working in the industry straight away. While thesis and projects do help students learn new things in the process, they still would not know how the industry is like or how it is like to work in an office environment. A lot of students finish university and do not know what they are going to do, and what skills they should learn in order to get a job. If they do an internship in BRAC beforehand, they will already have the experience required for them to get a job and they will also have a clear idea on what it is that they should learn.

Chapter 8

Conclusion

It has been more than 6 months since I have started my internship program for CSE400 and I have maintained to stay in Aveneur Solutions during this. With a heavy heart, I must say that my internship at Aveneur has come to an end.

I have joined as an intern in Aveneur Solutions – a small and new Software Development company which follows a startup culture. I joined there so that I could learn more about the IT industry and because I wanted to be around people who have a passion for what they do. Aveneur Solutions always targets aesthetics first as they want the look of their products to be on par with modern websites. It is a small office located in Rampura, and it is in the Senoark Building. The company recruits its developers through a series of examinations and interviews, and by checking candidates' portfolios. The company itself has only one branch for now, but wishes to expand in the future.

The software development architecture followed in Aveneur is the MVC architecture where the database, the business logic and the UI are completely separated. It complements the RSTAPI architecture we follow as well. The Software Development Methodology practiced in Aveneur Solutions is the Agile Development Methodology, where collaboration is key and the product is developed in increments.

The projects I have worked on are Senoark & Aveneur's portfolio website, Your Lawyer App, Shawn Novel and Associates portfolio website, and Homeland. I have provided the description, requirement analysis, and time distribution of each project. I have also written about the parts of the projects I have worked on, complete with screenshots. For the main project I have added a project break down structure as well. The most commonly used project tools and packages we used in the projects were also provided as well.

Out of the projects that I have written about, I have chosen my main project to be the e-commerce web application for Homeland. It is essentially a platform where users can book flats available within the Homeland Residential Society. Through this project, they wish to give their customers an inside look as to what their modern Society will be like for people targeted in the lower-middle class. Though this project was not created with scalability in mind, it would require a lot of concurrent users to use the website, so problems should not arise.

The training process started off with me doing small things like making buttons, containers, and layouts in HTML and CSS. Then I was given a course to watch online which comprised of how to build a project using React TypeScript and .Net core. I also learned a lot about how APIs work from my fellow developers. From the course I have learned that React allows easy creation of dynamic website, due to it being a declarative language and it having features such as creation of components and component wise refreshing instead of refreshing the whole page when it detects a change to be made. I have also learned how to use the Semantic UI library for react, so that I can use small and common components such as buttons and loaders which are already pre-made. As for .Net Core, I was told not to look too much into it due to time constraints, so I just learned about how it works and how APIs are created and used to connect the backend and the frontend part of the project. APIs are mainly microservices that take a request with JSON format and sends back a customized response from that data. They act as the intermediary between the database and the UI. We have also used a lot of tools like Adobe XD, VS Code, Github, ClickUp, Postman and Discord for a various reason such as UI/UX creation, Coding, Version Control, Making To-Do Lists, Testing APIs and primarily communication.

In this report, I have also talked about my team members and how the team is usually assembled for each project. I have written about how we partake in a lot of activities that improve teamwork and how we keep our morale high by celebrating and participating in company events.

Near the end, I have written about how this Internship Program has benefitted me and allowed me to grow. I talked about my professional growth where I experienced the challenges of adapting to both office and work environment, being punctual, maintaining deadlines and improving my communication skills. It was by overcoming those challenges that I was able to improve so much. In the same way I talked about my growth in a more technical sense. I talked about how I increased my proficiency in JavaScript, Frameworks and also learning many was to approach a specific problem.

In the end, I talked about what the impression I had on the Internship Program and what improvements could be made from both Aveneur's side and BRACU's side.

In conclusion, journey in Aveneur was long but well spent, but as we all know, all things must come to an end. I completed my journey with a smile on my face, and would recommend people to seek out Aveneur whether you are looking work there, or whether you want their services. It is a company filled with friendly faces who will make you feel right at home.

Bibliography

- [1] Aveneur-Solutions, “About aveneur,” 2020. eprint: <https://www.aveneur.com/#/about>. [Online]. Available: <https://www.aveneur.com/#/about>.