

Industrial Attachment (Sessional)

Course Code: CSE-402



Industrial Attachment

Brain Station 23

**Department of Computer Science and
Engineering**

Chittagong University of Engineering and Technology

16 October, 2023

Supervised By

Name	Designation
Dr. A. H. M. Ashfak Habib	Head & Professor, Dept. of CSE, CUET
Sharmistha Chanda Tista	Assistant Professor, Dept. of CSE, CUET
Hasan Murad	Lecturer, Dept. of CSE, CUET
Saadman Sakib	Lecturer, Dept. of CSE, CUET

Instructors' Information

Name	Designation
S. M. Siyam Hossain	Senior Executive, HR, Brain Station 23
Md. Mohoiminule Islam Chowdhury	Project Manager, Brain Station 23

Team Members' Information

Name	ID
Zannatul Fardaush Tripty	1804030
Sadia Ahmed	1804040
Estiak Ahamed Sazid	1804051
Karnis Fatema	1804052
Tahlil Abrar	1804056
Zerin Shaima Meem	1804057
Md. Arian Al Nafis	1804111
Omar Farook	1804125

Contents

1	Introduction	1
1.1	Vision	1
1.2	Mission	1
1.3	Services Provided By Brain Station 23	2
2	Learning and Assignments	5
2.1	Software Development Phases	5
2.2	Agile Software Modeling	6
2.3	SCRUM Meeting	7
2.4	Version Control Tool - GitHub	8
2.5	Accomplised Assignments & Project	9
2.5.1	HTML	9
2.5.2	CSS	10
2.5.3	Bootstrap	10
2.5.4	Tailwind CSS	11
2.5.5	Javascript	11
2.5.6	React JS	12
2.5.7	Overview of project	13
3	Hardware & Software Requirements	14
3.1	Hardware Requirements	14
3.2	Software Requirements	14
4	Project Description	15
4.1	Introduction	15
4.2	Frontend	16
4.3	Use of Hooks	24
4.4	Manage Routes	25
4.5	JSON Server	26
4.5.1	Uses of JSON Sever	26
4.5.2	Accessing Mock API	27
5	Conclusion	30

List of Figures

2.1	Phases of Agile Software Modeling	6
4.1	Flowchart for Frontend Development	16
4.2	Landing Page	16
4.3	Sign In As Admin	17
4.4	Register As Employee	17
4.5	Sign In As Employee	18
4.6	Register As Employee	18
4.7	Navigation Bar	19
4.8	Side Navigation Bar	19
4.9	Dashboard Part 1	20
4.10	Dashboard Part 2	20
4.11	Employee List	21
4.12	Add Employee Form	21
4.13	Profile Part 1	22
4.14	Profile Part 2	22
4.15	Edit Profile Form	23
4.16	Projects	23
4.17	Footer	24
4.18	use of useState Hook	24
4.19	use of useEffect Hook	25
4.20	React Routs	25
4.21	Nested Routing in App Component	26
4.22	Signup and POST method using JSON Server	27
4.23	Singing and fetching data from JSON Server	28
4.24	package.json file configeration of JSON server	28
4.25	Saving user Identity as object format	29
4.26	Saving current projects as array of object format	29

Chapter 1

Introduction

About Brain Station 23

Brain Station 23, founded by Raisul Kabir in 2006, stands as a prominent software development company in Bangladesh. Initially the company focused on the global market, the local market was introduced in 2010. Their expertise encompasses custom software, web and mobile app development, cloud solutions, and e-commerce platforms, serving a wide range of industries. With consistent growth, the company now boasts a team of over 700 software engineers, establishing its presence not only in Bangladesh but also in countries like the USA, UK, Netherlands, Denmark, Japan, Norway, and Sweden. They completed more than 2000 projects. It has over 12 tech partners including Amazon Web Services. It has served in more than 25 countries.

1.1 Vision

The vision of the Brain Station 23 is to be the fastest digital transformation and innovation partner by engaging global talents thus creating positive impact. They intend to be the customer-centred company. They emphasize on Leadership values such as ownership to take responsibility and to own the challenges and Honesty is also a significant part of their vision where they believe to say what one thinks and to do what one says. Team spirit is also one of their core principles. They believe to be successful at work, one must be passionate and committed to it. Their main belief is that to deliver best solution one has to stay agile.

1.2 Mission

Their goals are

- To connect with people for digital leadership by empowering people to achieve more with less.

- To empower people with IT industry and grow as a nation.
- To serve Enterprises and SMEs with technological partnership.
- To emphasize on latest technologies for digital transformation.
- To be peoples innovation and success partner regardless of diverse culture.

1.3 Services Provided By Brain Station 23

- **Web App Development**

Brain Station 23 offers a comprehensive range of secure software solutions for banking businesses and financial institutions. Their services encompass retail and corporate internet banking, mobile internet banking, card services, extensive customization, FDI automation, internet banking systems, cyber security and audit, banking process automation, and banking CRM.

- **Mobile App Development**

They offer top-notch mobile application development services, renowned for their capacity to translate solutions into tangible business growth. They specialize in creating cross-platform, responsive mobile applications, prioritizing appealing user interfaces and secured backends. They cater to a wide range of industries and clients, including both large-scale organizations and Uber-like startups.

- **AEM Solutions**

Brain Station 23 offers AEM or Adobe Experience Manager development with implementation, integration, managed service along with intense customization for AEM platform. Brain Station 23 also offers migration and upgrade with the help of experts hand so that one do not have to take any hassle.

- **ML & AI**

They offer below mentioned services of projects related to Artificial Intelligence and Machine Learning.

- Customer/Human Behavior Analysis.
- Face Identification.
- Process Automation.
- Pattern Analysis.
- Intelligent Chatbot.
- Fraudulent Identification.
- Data Analysis for Biological Macromolecular Structures
- Predictive Analytics Modeling.

- Targeted Marketing.
- Churn Analysis.
- Trend Analysis.
- Time Series Analysis.

- **Cloud Solution**

Brain Station 23 is a highly regarded Global Advanced AWS Consulting Partner. They specialize in providing cloud consulting, adoption, and management services to a diverse range of businesses, including Enterprises, SMBs, and startups. The company takes pride in being fast, precise, innovative, and flexible in their approach as a cloud partner. They offer solutions with constant 24/7 support and monitoring. Brain Station 23 delivers complete cloud solutions customized for different industry sectors.

- **E-Commerce**

They provide a top-notch e-commerce solution, customized for individual businesses. They focus on improving functionality and performance, tailoring services to specific needs. With their team's expertise, both business owners and customers can expect a smooth and optimized e-commerce experience.

- **BI Solution**

They offer a comprehensive data management solution, covering everything from creation to storage and archiving. The certified experts are dedicated to providing modern and cost-effective solutions, maximizing the value of individuals' data. They prioritize data accuracy, completeness, consistency, and accessibility, ensuring trustworthy information.

- **ERP**

It is the stepping stone for automation and here company manage key parts of operations including resource management and accounting. The products which are mentioned below are made by Brain Station 23.

- **HRMS:**

It is used to bring HR functionalities under one platform with HRMS solutions.

- **Pharma POS:**

It brings all Pharmaceutical stores and counters under one platform.

- **People Tracker:**

It is a GPS tracking solution to bring best productivity for remote employees.

- **Freight Forward and Logistics:**

It allows individuals and companies to control and manage booking orders, shipping orders, shipment advice, bill of landing, consignment, track daily update.

- Accounting:

It automates ones financial operations. It streamlines ones financial processes, generate reports, set budgets.

• E-Learning

Brain Station 23 developed LMS platform **eLearning 23** with Moodle, to create the online learning environment the way people imagined.

• AR VR Solution

- Augmented Reality:

They add digital features to the real environment. We make augmented reality apps for iOS and Android. They use technologies such as vuforia, maxst, ARCore.

- Virtual Reality:

They use VR development services to improve business appearance by translating ideas to reality. They also provide unique VR services to meet business demands and to turn ideas to cutting edge technology.

• SharePoint:

Brain Station 23 implements SharePoint solution based on your specific needs with rapid deployment to faster the Return On Investment for ones business. The provided services are.

- Enterprise Portal Development.
- SharePoint Application Development.
- Business Application.
- Customization and Enhancement in Branding.
- Performance Tuning.
- Migration and Upgrades.

They have worked with Robi, UCB, Beximco Pharmaceuticals and Berger also.

• Software Testing As A Service

Brain Station 23 Ltd specializes in simulating real-world testing environments and finding bugs in the software product. By STAS one can have a dedicated trained QA team working on the project On-demand with dynamic assets and sanitized test data. The skills or resources to carry out automation or manual testing can be intensive and varies project to project. Here, They ensure the best approach to mitigate the client's demand for software quality with relevant testing activities as per SDLC and performing at best.

Chapter 2

Learning and Assignments

2.1 Software Development Phases

Software development typically involves several phases or stages, each of which contributes to the creation of a software product. The exact phases may vary depending on the software development methodology being used (e.g., Waterfall, Agile, DevOps). Here are the common phases in the software development lifecycle:

- **Planning:** In this initial phase, the project's goals, scope, and requirements are defined. This phase involves setting objectives, conducting feasibility studies, and creating a project plan.
- **Analysis:** During this phase, the project team, including business analysts and developers, works to understand and document the detailed requirements of the software. This involves gathering information from stakeholders and creating use cases, flowcharts, and other documentation.
- **Design:** In this phase, the software's architecture and design are created based on the requirements gathered in the analysis phase. This includes defining data structures, algorithms, and system interfaces.
- **Implementation (Coding):** Developers write the actual code based on the software design. This is where the software is built and tested.
- **Testing:** Once the software is tested and approved, it is deployed to a production environment, making it available to users. Deployment may involve setting up servers, databases, and other infrastructure.
- **Maintenance and Support:** After deployment, the software enters a phase of ongoing maintenance and support. This includes fixing bugs, making updates, and providing technical support to users.

- **Documentation:** Throughout the software development lifecycle, documentation is created to help developers, users, and other stakeholders understand the software's operation and maintenance.
- **Quality Assurance and Quality Control:** Quality assurance ensures that the software development process follows established standards and best practices. Quality control focuses on testing the software and ensuring that it meets quality standards.
- **Feedback and Iteration (Agile Methodologies):** In Agile methodologies, including Scrum and Kanban, software development occurs in iterative cycles. After each iteration or sprint, feedback is collected, and the product is improved based on this feedback. This process is repeated until the software is considered complete.
- **Release and Continuous Deployment (DevOps):** In DevOps, the emphasis is on automating the deployment process, enabling continuous integration and continuous delivery (CI/CD) pipelines. Software is continuously deployed to production, and updates are made rapidly in response to user feedback.

2.2 Agile Software Modeling

The Agile model is an iterative and incremental approach to software development that focuses on flexibility, collaboration, and customer satisfaction. It doesn't have strict phases in the traditional sense, but it does have key activities and principles that guide the development process.



Figure 2.1: Phases of Agile Software Modeling

- **Requirements Gathering:** Agile places a strong emphasis on understanding and prioritizing customer needs. During this phase, a product backlog is created, which is a dynamic list of features, user stories, or tasks that need to be addressed in the software.
- **Design the Requirements:** Agile design is often an ongoing and parallel process to development. Rather than doing all the design upfront, teams may design elements as they work on them. This approach allows for flexibility and responsiveness to changing requirements.
- **Construction/Iteration:** In Agile, development work is organized into iterations or sprints. Each iteration is typically a short, fixed period (e.g., 2-4 weeks) during which a specific set of user stories or features are developed. The construction phase involves building, coding, and testing these features incrementally.
- **Testing/Quality Assurance:** Testing is an integral part of each iteration. Developers and testers work collaboratively to ensure that new features are thoroughly tested and meet the acceptance criteria defined in the user stories. Automated testing is often used to maintain the quality of the software.
- **Deployment:** Agile allows for frequent, incremental deployments. After each sprint or iteration, a potentially shippable increment of the product is produced. It may not be released to customers immediately, but it's ready for deployment when the time is right.
- **Feedback:** Agile encourages regular feedback from stakeholders, including customers and end-users. It's important to gather input on the incrementally developed features, as this feedback informs future iterations and helps refine the product.

2.3 SCRUM Meeting

Scrum is an agile framework that teams use to produce products faster by breaking large development projects into smaller pieces that can be completed in short timeframes. Scrum meeting is a catch-all term that can describe different types of meetings held by Scrum teams. Scrum meetings include daily standups, sprint planning sessions, and sprint retrospectives. A Scrum meeting can refer to any meeting held by a Scrum agile team during a product's development. Here are the most common types.

- **Daily Scrum:** The daily scrum, also called the standup, is a short daily meeting designed to let the team plan out its work for the day and identify any obstacles that could impact that work.

- **Sprint Planning:** It is a team meeting held before the next agile sprint. The team reviews its backlog during sprint planning and decides what items to prioritize for the next sprint.
- **Sprint Retrospective:** It is a post-sprint discussion. The Scrum team reviews what happened during the sprint to determine what worked, what didn't work, and how they can improve the process during the next sprint.

2.4 Version Control Tool - GitHub

GitHub provides a wide range of services and features related to version control, collaboration, and software development. Here are some of the key offerings and benefits that GitHub provides:

- **Version Control:** GitHub is primarily known for its Git-based version control system. It allows us to track changes in your codebase, collaborate with others, and manage different versions of your software.
- **Code Hosting:** We can host our code repositories on GitHub, making them accessible to collaborators and the public. This is often used for open source projects and private repositories for commercial software.
- **Collaboration tool:** GitHub provides collaboration features such as pull requests, issues, and code reviews that make it easy for developers to work together, discuss changes, and merge code.
- **Project management:** GitHub includes project boards and issue tracking to help teams organize their work, prioritize tasks, and track progress.
- **Code hosting with documentation:** Besides code, GitHub can host documentation, wikis, and other project-related files. This makes it a versatile platform for both code and project management.
- **Community and Social Features:** GitHub has a strong social aspect. We can follow other developers, explore open-source projects, and contribute to projects you find interesting.
- **Security Features:** GitHub provides security scanning for vulnerabilities in our code, dependency tracking, and alerts for known security issues.
- **GitHub Pages:** We can use GitHub Pages to host a website directly from your GitHub repository, making it easy to showcase your projects, documentation, or personal websites.

- **Education and Learning:** GitHub provides resources for students and teachers to learn and teach programming, making it a valuable platform for educational purposes.
- **Mobile Apps:** GitHub offers mobile apps for iOS and Android, allowing us to manage our repositories and engage with the GitHub community from our mobile device.
- **APIs:** GitHub offers a comprehensive API that allows developers to integrate GitHub into their workflows and applications.

2.5 Accomplised Assignments & Project

2.5.1 HTML

HTML (Hypertext Markup Language) is a standard markup language used to structure and create web content. It consists of elements and tags that define the structure and semantics of a webpage, including headings, paragraphs, links, images, and more. Browsers use HTML to render and display web pages to users.

- **Structured Content:** HTML provides a structured way to organize content on webpages, using elements like headings, paragraphs, lists, and tables. This structured approach enhances the clarity and organization of information.
- **Hyperlinking:** HTML enables the creation of hyperlinks (links) that connect webpages and resources, facilitating navigation within websites and across the internet.
- **Cross-Platform Compatibility:** HTML is a universal standard understood by web browsers on various devices and operating systems, ensuring consistent rendering of web content.
- **Integration with Other Technologies:** HTML can be combined with CSS for styling and JavaScript for interactivity, creating rich and dynamic web experiences.
- **Wide Adoption:** HTML is widely adopted and supported, making it a fundamental tool for creating web content, from basic text documents to complex web applications.
- **Responsive Web Design:** HTML is a key component in creating responsive web design, allowing web developers to design web content that adapts to different screen sizes and devices.

2.5.2 CSS

CSS (Cascading Style Sheets) is a stylesheet language used in web development to control the visual presentation of HTML elements. It allows for the definition of styles like colors, fonts, layouts, and animations, enabling designers to customize the appearance of web pages. Some of the important features of CSS' are mentioned below:

- **Separation of Content and Presentation:** CSS allows for the separation of the content (HTML) and its presentation (styling). This separation makes it easier to update and maintain websites because changes to the visual style can be made without altering the content itself.
- **Consistent Styling:** CSS enables the application of consistent styling across a website or web application. You can define styles for various elements and ensure they are applied uniformly throughout the site.
- **Responsive Design:** CSS supports responsive design by allowing the creation of styles for different screen sizes. Media queries in CSS enable the adaptation of layouts and styles for various devices, such as mobile phones, tablets, and desktops.
- **Typography Control:** CSS offers precise control over typography, including font choices, sizes, line spacing, and more.
- **Layout Control:** CSS provides a variety of tools for controlling the layout of web pages, including properties like 'display', 'position', and 'float'. This enables the creation of complex and responsive page structures.
- **Animation and Transition Effects:** CSS enables the creation of animations and transitions to enhance user interactions and make websites more engaging. Properties like 'transition' and 'animation' can be used to achieve these effects.
- **Cross-Browser Compatibility:** CSS helps ensure that web pages are displayed consistently across different web browsers, allowing web designers and developers to write code that works on various platforms.

2.5.3 Bootstrap

Bootstrap is a popular front-end framework for building responsive and visually appealing web applications. It offers a variety of components and styles that can be particularly useful when working with React applications. Some of the benefits of bootstrap are:

- It provides us a responsive grid system that helps in creating flexible and responsive layouts. This is crucial for building web applications that adapt to various screen sizes and devices.

- Bootstrap allows for customization of its styles. We can use Bootstrap's default styles or modify them to match your application's design.
- By using Bootstrap components, we can maintain a consistent look and feel throughout your React application.
- Bootstrap allows us to create custom themes to match your brand or design requirements.
- Bootstrap can significantly speed up the development process for building user interfaces in React.

2.5.4 Tailwind CSS

Tailwind CSS, a popular utility-first CSS framework, provides a range of advantages for web developers and designers. The benefits that tailwind css provides us are:

- **Utility-First Approach:** Tailwind CSS promotes a utility-first approach to styling, which means that we create our designs by applying small, single-purpose utility classes directly to HTML elements.
- **Modularity:** Tailwind is highly modular, allowing users to selectively employ the CSS classes they require, thereby ensuring a lean and efficient codebase.
- **Responsive Design:** Tailwind includes responsive classes that simplify the creation of responsive designs for different screen sizes. This eliminates the need for writing custom media queries.
- **Customization:** Tailwind is exceptionally customizable. Users can configure the framework to generate CSS that aligns with their project's specific requirements.
- **Pre-Built Components:** Tailwind features a collection of pre-designed components, such as buttons, cards, and navigation bars, which can be easily tailored to match a project's unique design.
- **Scalability:** Tailwind is well-suited for both small-scale projects and extensive applications. Its modular structure and organization make it adaptable for projects of all sizes.
- **Documentation:** Tailwind offers comprehensive and well-structured documentation that simplifies the learning and effective utilization of the framework.

2.5.5 Javascript

Firstly we were given a plalylist to learn basic JavaScript. Each members of the team finished every videos of the playlist. We learnt:

- Fat arrow function (`=>`): A concise way to define functions in JavaScript that retains the context of the surrounding code (lexical scoping).
- `this` keyword: Refers to the current object or context and can be affected by how a function is called, often leading to unexpected behavior in JavaScript.
- Truthy/Falsy values: In JavaScript, values that evaluate to true or false in a boolean context; truthy values are treated as true, while falsy values are treated as false.
- **Ternary operators (conditional operator):** A concise way to write conditional statements in the form of `condition ? trueValue : falseValue`.
- **Array functions :** `find()`, `findIndex()`, `filter()`, `slice()`, `splice()`, `concat()`, `push()`, `reduce()` Methods for manipulating arrays in JavaScript, offering various operations like searching, filtering, slicing, modifying, and reducing array elements.
- **Loops (for-in, for-of):** Iteration structures in JavaScript; "for-in" iterates over object properties, while "for-of" is used to iterate over iterable values (e.g., arrays, strings).
- **Spread operator:** A JavaScript feature (e.g., `'...'`) used for expanding array elements or object properties in function arguments or array literals.
- **Rest operator:** An operator in JavaScript (usually denoted as `...`) used to capture multiple function arguments into a single array.
- **Object destructuring:** A way to extract values from objects and assign them to variables using a shorthand syntax.
- **Template literals:** A string notation in JavaScript, denoted with backticks `(`)`, that allows the embedding of variables and expressions within the string.

2.5.6 React JS

React is an open-source JavaScript library for building user interfaces, emphasizing component-based development and efficient updates to the user interface. It is widely used for creating dynamic and interactive web applications.

- **Component-Based Architecture:** React promotes a modular and component-based approach, making it easier to manage and reuse UI elements, leading to more maintainable code.
- **Virtual DOM:** React uses a virtual representation of the DOM, which improves performance by minimizing direct interaction with the actual DOM, resulting in faster updates and a more responsive user interface.

- **Efficient Updates:** React efficiently updates only the parts of the UI that have changed, reducing the need for extensive manual DOM manipulation and improving application performance.
- **Large Ecosystem:** React has a large and active ecosystem with a wealth of libraries, tools, and community support, making it easier to integrate with other technologies and solve common development challenges.
- **Server-Side Rendering (SSR):** React supports server-side rendering, which improves SEO and initial page load times by rendering content on the server before sending it to the client.

2.5.7 Overview of project

After our supervisor Sadmaan Sakib sir consulted with the HR as well as project manager, they assigned us a project. The project was about task management for the admin of an organization. We were told that the whole project will be done with React Js. We had primary knowledge about HTML, CSS and JavaScript. It became easy for us to follow the documentation of react. Then we planned how the UI will look like, what functions there will be. The we divided our works into 4 sections. Each sections were assigned to 2 members of the team. We had to learn how to divide the whole project into components, the how to reuse each components, the standard folder structure of react. We also had to learn some CSS frameworks which will be discussed afterwards.

Chapter 3

Hardware & Software Requirements

3.1 Hardware Requirements

In our project, we used the following hardware:

- Personal Computer
- Hard Disk
- Ram - 8 GB or more

3.2 Software Requirements

- OS: Windows 10 or above
- Languages: HTML, CSS, JavaScript
- Library: React JS
- Framework: Bootstrap, Tailwind CSS
- IDE: VS Code
- Shell: Windows Command Prompt(cmd), GIT Bash
- Browser: Google Chrome

Chapter 4

Project Description

We have developed a project, named "Employee Management System", which is a web application, designed for Brain Station 23. It allows administrators and employees to create accounts, sign in, and manage projects. Administrators can assign and track projects, view their own projects, and edit their profiles. This system enhances project management, transparency, and communication within the organization. This project is mainly focused in frontend of the application, which is driven by JSON Server.

4.1 Introduction

The full project is completed by using a frontend library React JS. The user interface is designed with the help of CSS, Bootstrap, and Tailwind CSS.

First, we have created the required react components for our projects. React components are basically built with JSX. JSON Server is used as a mock REST API to perform some tasks such as sign-in and sign-up. We have also used JSON Server for using some dummy data in our projects.

By using the concepts of hooks of react js such as useState, useEffect we have performed our required tasks for the projects. React router is used to manage the routes in web application.

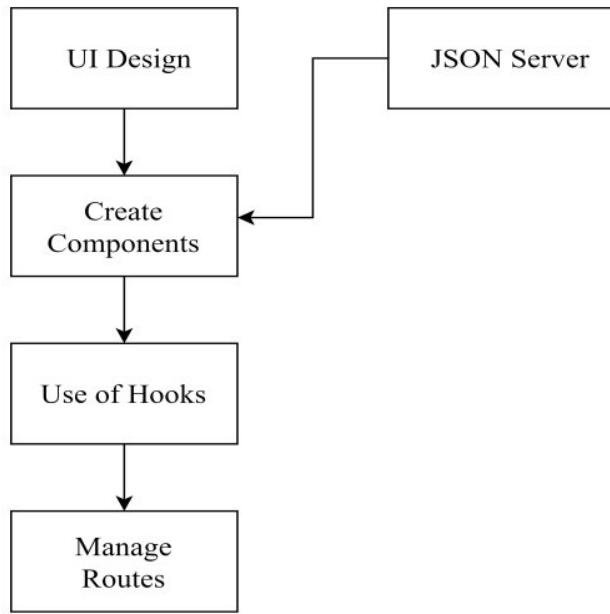


Figure 4.1: Flowchart for Frontend Development

4.2 Frontend

- Landing Page

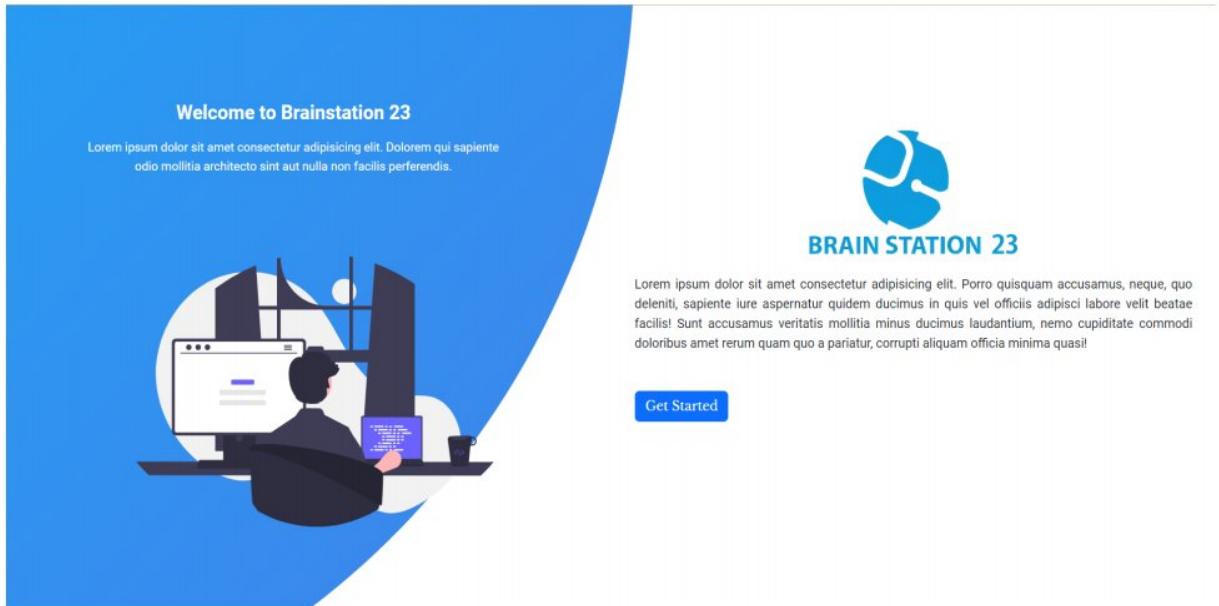


Figure 4.2: Landing Page

- Sign In As Admin

After the user presses get started from the previous page, he can see a page with the title Sign in as Admin. On the left side, he can also see the option to sign in as an employee. The page will toggle on the choice of admin or employee.

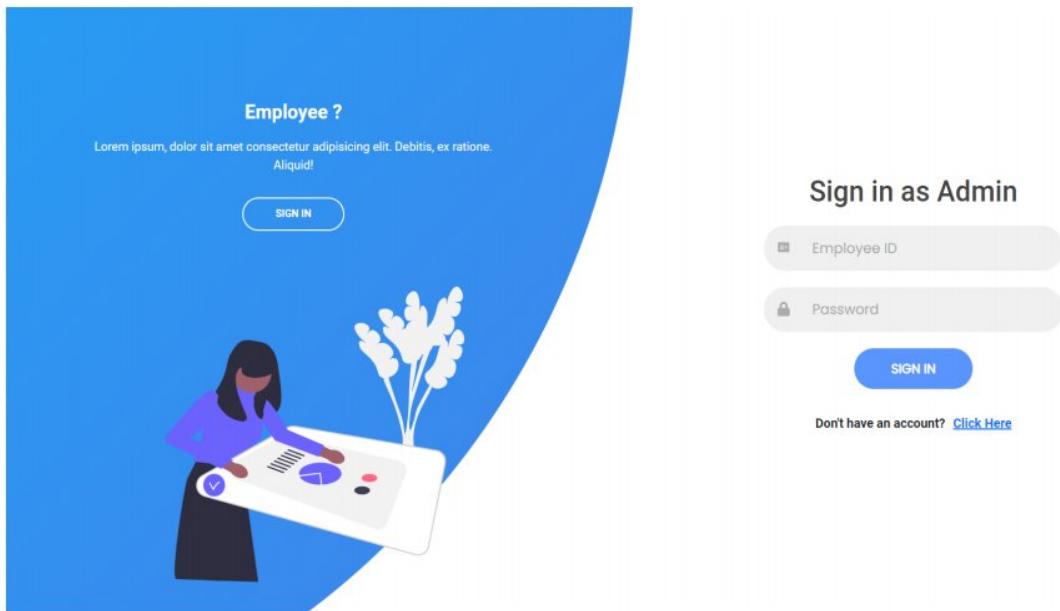


Figure 4.3: Sign In As Admin

- **Register As Admin**

In this page, if an admin is not already signed in, then he/she can register by providing name, last name, first name, email and password.

Figure 4.4: Register As Employee

- **Sign In As Employee**

In this page, employee can sign in using his Employee ID and password and at the right side, there is a sign in option for the Admin.

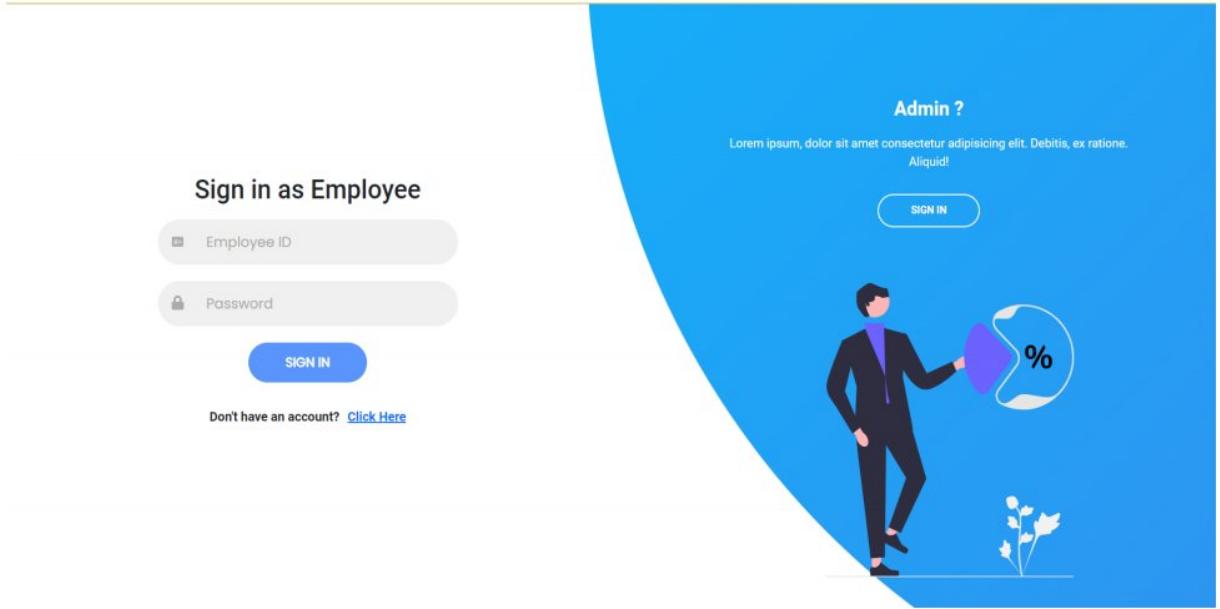


Figure 4.5: Sign In As Employee

- **Register As Employee**

In this page, if an employee is not already signed in, then he/she can register by providing employee name, last name, first name, email and password.

The image shows the 'Register as Employee' page. It has a light blue header with the text 'Register as Employee'. Below the header are five input fields: 'Employee ID', 'First Name', 'Last Name', 'Email', and 'Password'. A blue 'SIGN UP' button is located at the bottom of these fields. At the bottom left, there is a link 'Already have an account? [Click Here](#)'. On the right side of the page, there is a logo for 'BRAIN STATION 23' featuring a stylized brain icon above the text 'BRAIN STATION 23'. Below the logo is a block of placeholder text: 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Porro quisquam accusamus, neque, quo deleniti, sapiente iure aspernatur quidem ducimus in quis vel officiis adipisci labore velit beatae facilis! Sunt accusamus veritatis mollitia minus ducimus laudantium, nemo cupiditate commodi doloribus amet rerum quam quo a pariatur, corrupti aliquam officia minima quasi!'

Figure 4.6: Register As Employee

- **Navigation Bar**

The navigation bar has a logo of the company, a search bar, a profile icon, and a signout icon. Search bar is used fo search by Employee name or Project name. Using the profile icon user can navigate to his own profile. And by clicking on signout icon, one can immidiately signout from the page.

Figure 4.7: Navigation Bar

- **Side Navigation Bar**

The side navigation bar offers quick access to essential features. It includes a comprehensive dashboard for a holistic view of office operations, a profile section for managing personal information, an employee list for easy access to employees' contact details, and a project section for tracking and managing ongoing projects.

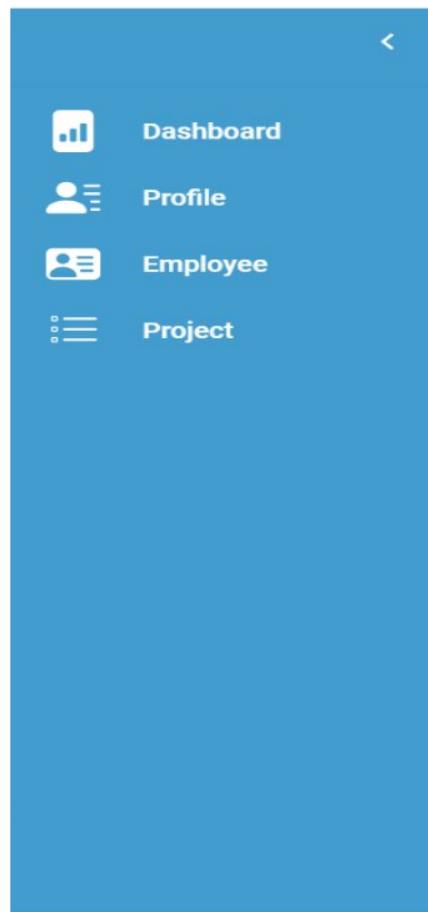


Figure 4.8: Side Navigation Bar

- **Dashboard**

The dashboard gives a complete overview of office activities. It begins with a line graph showing how many new employees presented in every year, followed by a pie chart illustrates the comparison between male and female employees, providing a visual representation of their annual presence in the office. The Employee availability section provides everyday details like being on time, absences, and leave requests. In the top right, recruitment numbers are shown for applicants, interviews, and hires. The bottom right component outlines the meeting schedules for interviewees,

ensuring a well-organized and efficient recruitment process. Most bottom bar chart depicts the number of developers joining in each sector every month of the year.

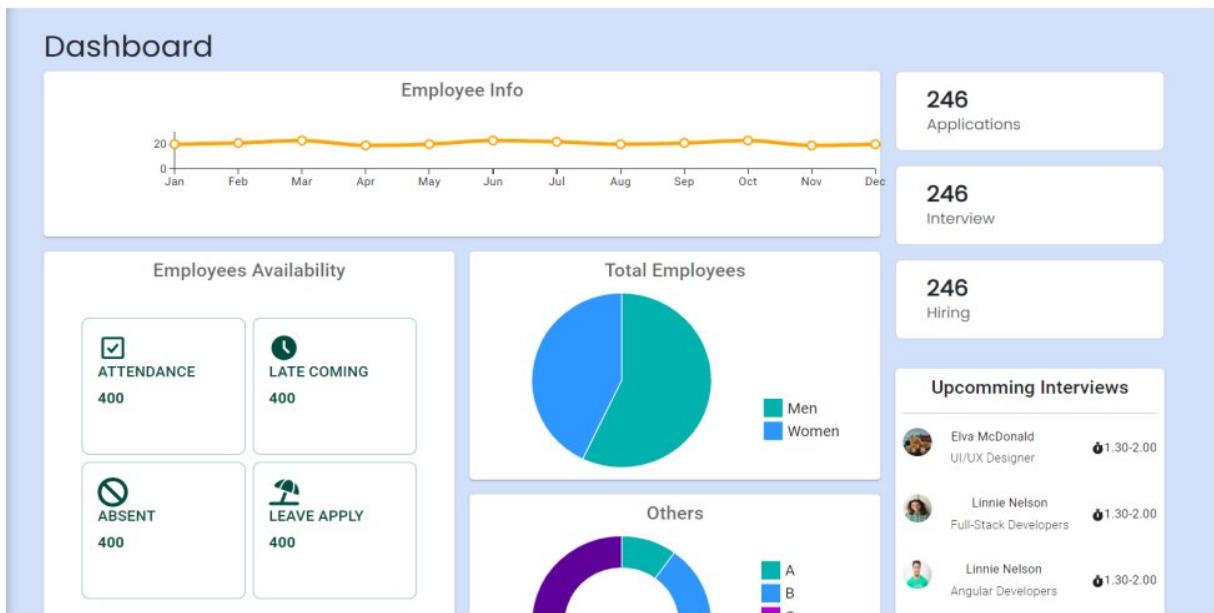


Figure 4.9: Dashboard Part 1



Figure 4.10: Dashboard Part 2

• Employee List

The employee list page displays employee details, including performance ratings based on stars. It also shows the number of projects they're completed.

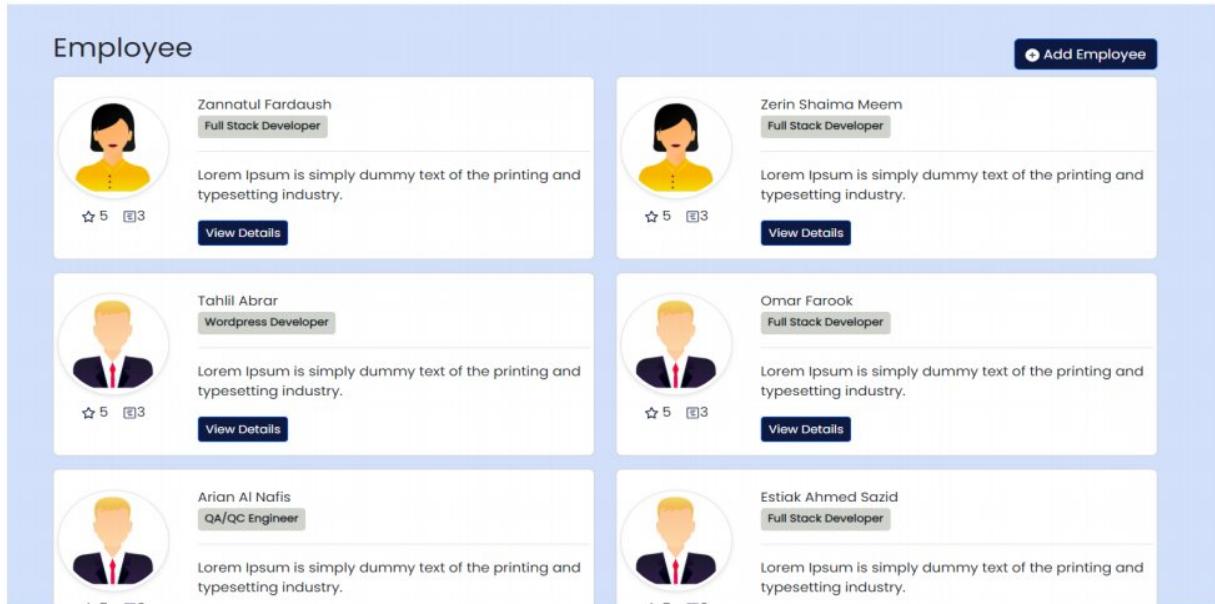


Figure 4.11: Employee List

In the employee's profile, you can find specifics about the projects they're working on.

• Add Employee

In this part, admin can create a new profile for a new employee that joined, with credentials provided in the form, a new profile is created.

The screenshot shows a modal dialog titled "Add Employee" with a close button at the top right. The form contains the following fields:

- Employee Name:** A text input field with placeholder "Enter Employee Name".
- Employee Email Id:** A text input field with placeholder "Email@gmail.com".
- Password:** A text input field with placeholder "Password".
- Department:** A dropdown menu set to "Web Development".
- Designation:** A dropdown menu set to "UI/UX Design".
- Employee ID:** A text input field with placeholder "12345".
- Joining Date:** A date input field with placeholder "mm/dd/yyyy".
- Description (Optional):** A text area for additional notes.

At the bottom right of the modal are two buttons: "Close" and "Save changes". In the background, the main employee list interface is visible, showing other employees like Zannatul Fardaush, Tahil Abrar, Arian Al Nafis, Zerin Shaima Meem, and Omar Farook.

Figure 4.12: Add Employee Form

- **Profile**

In this part, the admin can create a new profile for a new employee who joined, with credentials provided in the form, a new profile is created.

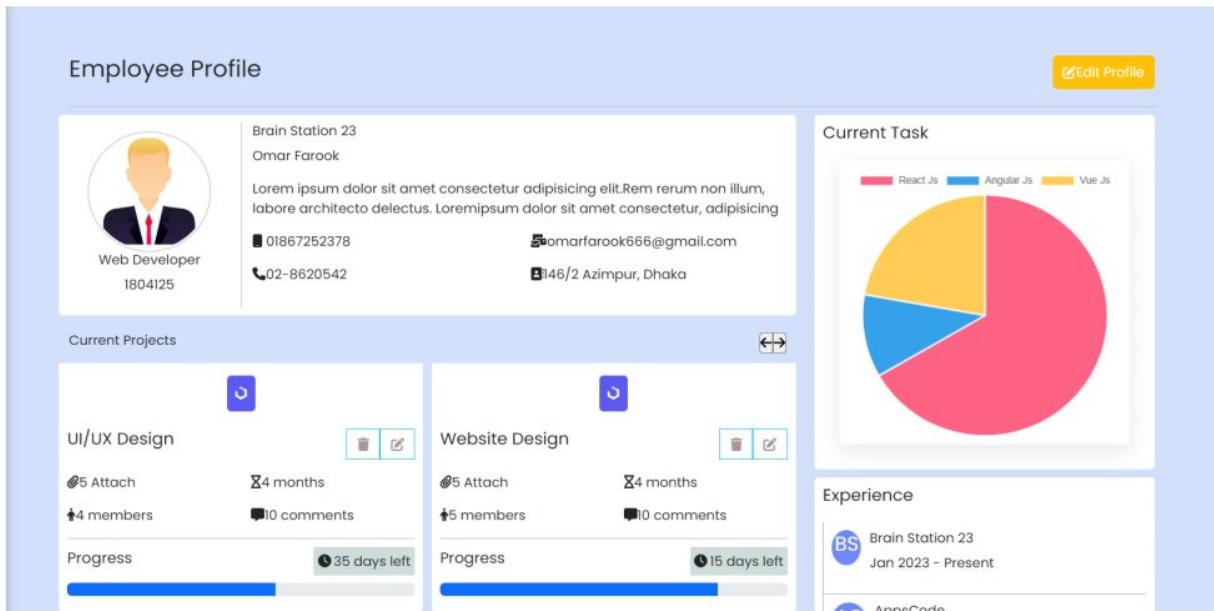


Figure 4.13: Profile Part 1

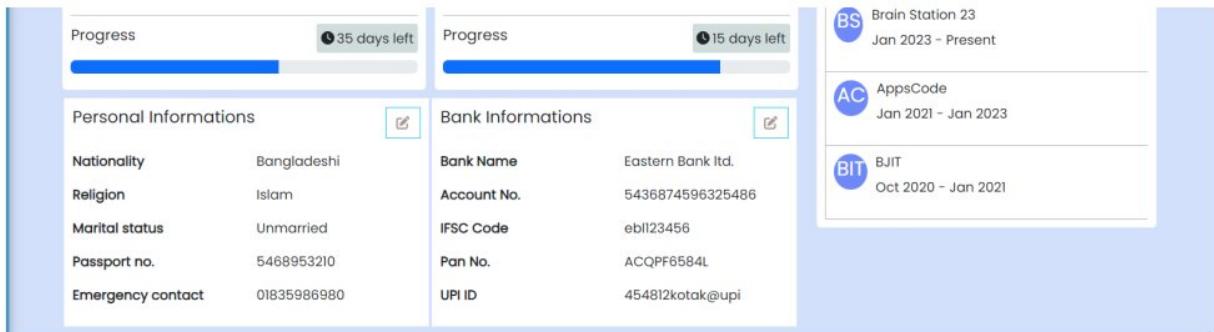


Figure 4.14: Profile Part 2

- **Editing Profile**

In this part, the admin can create a new profile for a new employee who joined, with credentials provided in the form, a new profile is created.

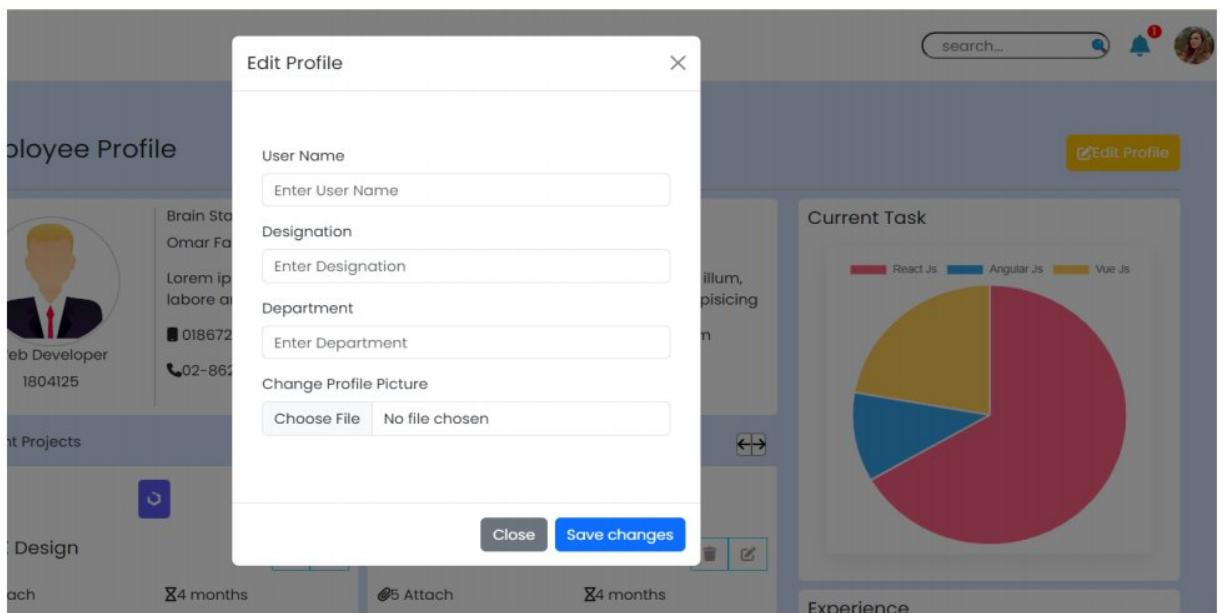


Figure 4.15: Edit Profile Form

- **Projects**

In this Projects page, Admin can manage tasks. Admin can create new projects, drag a project from the Todo list to Work in Progress or to Done column once the task is finished. If needed additional columns can be generated.

The screenshot shows a 'Task Manager' interface with three columns: 'Todo', 'Work in progress', and 'Done'. Each column has a header with a circular icon and a count (0). Below each header is a card containing a task description. At the bottom of each column is a blue button labeled '+ Add task'. To the right of the columns is a light blue area with a '+ Add Column' button.

Column	Tasks
Todo	List admin APIs for dashboard Develop user registration functionality with OTP delivered on SMS after email confirmation and phone number
Work in progress	Conduct security testing
Done	(Empty)

Figure 4.16: Projects

- **Footer**

The footer contains Address, Top trends and Contacts of our company Brain Station 23.

Brain Station23		Top Trends	Get In Touch
Serving Enterprises and SMEs with Technological Partnership Since 2006.	(Building-1) 8th Floor, 2 Bir Uttam AK Khandakar Road, Mohakhali C/A, Dhaka 1212, Bangladesh	Front-End Developers	Unity3D Developers
(Building-2) 4th Floor, 4 Bir Uttam AK Khandakar Road, Mohakhali C/A, Dhaka 1212, Bangladesh	Full-Stack Developer	iOS Developers	Tel: 02-222290728 Mob: +8801404055220 (Reception)
(Building-3) 3rd Floor, Mirpur DOHS Cultural Centre, Road 9, Mirpur DOHS, Dhaka 1216, Bangladesh	Angular Developers	Android Developers	Mob: +8801404055226 (Marketing & Sales) sales@brainstation-23.com
	Node.js Developers	nopCommerce Developers	career@brainstation-23.com (HR)
	JavaScript Developers	Drupal Developers	Mob: +8801404055227 (HR)
	React Developer	SQL Developers	
	.NET Developers	QA Engineers	
	Java Developers	UI/UX Designers	
	PHP Developers	Product Owner	
	Python Developers	Scrum Master	
	C/C++ Developers	DevOps Engineer	
	C# Developers		
	Ruby Developers		

Figure 4.17: Footer

4.3 Use of Hooks

In React.js, hooks are functions that allows us to "hook into" state and lifecycle features of class components without having to write a class. They provide a more concise and understandable way of working with state and side effects in functional components. Here are some React hooks, that we have used in our project:

useState: useState allows functional components to manage states. It takes an initial state as an argument and returns an array with the current state and a function to update it.

For example, we used useState in modal buttons, where button onClick, the boolean parameter basicmodal will toggle, and on basis of that, it will pop up the modal on the screen:

```
const [basicModal, setBasicModal] = useState(false);

const toggleShow = () => setBasicModal(!basicModal);
```

Figure 4.18: use of useState Hook

useEffect: useEffect is used for handling side effects in functional components. It takes two arguments: a function that contains the code you want to run as a side effect (this code will run after every render) and an array of dependencies that specify when the effect should run.

For example in our signin page, we used the concept of useEffect to handle scenarios where a user is automatically redirected to the dashboard they are already logged in.

```
useEffect(() => {
  let id = sessionStorage.getItem('id');
  if (id) {
    navigate('/App/Dashboard');
  }
});
```

Figure 4.19: use of useEffect Hook

4.4 Manage Routes

Routing is the process of managing and navigating between different views or pages within a single-page application (SPA). Routing allows us to create a multi-page-like user experience without the need to reload the entire web page when navigating between different sections of our application. React Router is a commonly used library for handling routing in React applications.

We used nested routing in our project. In the beginning, the site will take us to the landing page by default. Then on clicking Get Started, we can go to sign in or register pages. Once the user is signed in to the site, the App component is routed. Inside App component there are several pages like Dashboard, Employee list, Profile and Projects which can be navigated using the sidebar.

```
root.render(
  <React.StrictMode>
    <ToastContainer theme="colored" position="top-right"></ToastContainer>
    <BrowserRouter>
      <Routes>
        <Route exact path="/" element={<Home/>}/>
        <Route exact path="/App" element={<App/>}>
          <Route exact path="/App/Dashboard" element={<Dashboard/>}/>
          <Route exact path="/App/Project" element={<Project/>}/>
          <Route exact path="/App/Profile" element={<Profile/>}/>
          <Route exact path="/App/Employee" element={<Employee/>}/>
        </Route>
        <Route path="/Signin" element={<Signin/>} />
        <Route path="/Admin-sign-up" element={<AdminSignUp/>} />
        <Route path="/Emp-sign-up" element={<EmpSignUp/>} />
      </Routes>
    </BrowserRouter>
  </React.StrictMode>
);
```

Figure 4.20: React Routs

```
const App = () => {
  return (
    <div>
      <div>
        <HeadBar/>
        <Box sx={{display: 'flex', paddingTop:10}}>
          <SideBar/>
          <div>
            <Outlet/>
            <Footer/>
          </div>
        </Box>
      </div>
    </div>
  )
}
```

Figure 4.21: Nested Routing in App Component

4.5 JSON Server

JSON Server is a popular and simple-to-use tool that allows developers to set up a mock REST API server quickly. It is particularly useful during the development and testing phases of a web application. JSON Server creates a full-fledged REST API based on a JSON file or JavaScript object, which can be used to mimic the behavior of a real backend server without actually having one.

We have used JSON server as a mock REST API Server for our Employee Management System Project. Sign-up and Sign-in both are done by using the JSON Server. First, we have created the JSON Server at another port. Then, we used this as a backend to put our Sign up data both for admin and Employee. After that, we fetch the server from that specific port to signin in our dashboard. A successful fetching of data shows a success message and is redirected to the admin of the employee dashboard. A mismatch of information or unsuccessful fetching of data shows a failure message.

4.5.1 Uses of JSON Sever

- **Installation**

To get started, we needed to install JSON Server globally on our development machine using npm or yarn: `npm install -g json-server`.

- **Creating a JSON File**

After that, we have create a JSON file that contains our mock data. This file typically represents a collection of resources (e.g., admin, users or employees) as an array of JSON objects.

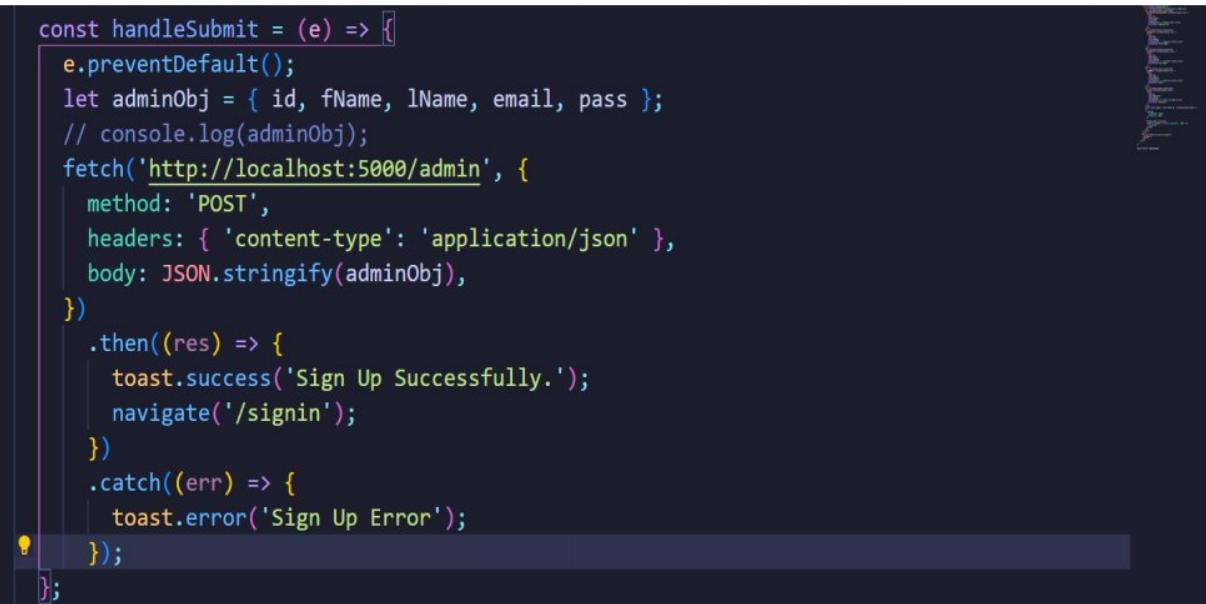
- **Running JSON Server**

To run the Server the following command is called: json-server -db.json -port 5000. This starts the JSON Server and provides you with a set of REST API endpoints based on the data in your JSON file. By default, it runs on port 5000, but you can specify a different port if needed.

4.5.2 Accessing Mock API

- **Sign-up**

For both admin and employee sign-up by sending a POST request to the custom sign-up endpoint (e.g., /admin).



```
const handleSubmit = (e) => [
  e.preventDefault();
  let adminObj = { id, fName, lName, email, pass };
  // console.log(adminObj);
  fetch('http://localhost:5000/admin', {
    method: 'POST',
    headers: { 'content-type': 'application/json' },
    body: JSON.stringify(adminObj),
  })
    .then((res) => {
      toast.success('Sign Up Successfully.');
      navigate('/signin');
    })
    .catch((err) => {
      toast.error('Sign Up Error');
    });
};
```

Figure 4.22: Signup and POST method using JSON Server

- **Sign-in**

For the sign-in the standard '/admin' or '/employee' is used to fetch the relevant data from port address 5000. We fetch the data according to the 'id' given by the user while signing. If the given data matches with fetched data than sign-in is successful otherwise an unsuccessful message is shown on the right top.

```

const handleSignIn = (e) => {
  e.preventDefault();
  fetch(`http://localhost:5000/admin/${id}`)
    .then((res) => {
      console.log(res);
      return res.json();
    })
    .then((resp) => {
      if (Object.keys(resp).length === 0) {
        toast.error('Please enter valid ID');
      } else {
        if (resp.pass === pass) {
          toast.success('Success');
          sessionStorage.setItem('id', id);
          navigate('/dashboard');
        } else {
          toast.error('Please enter valid credentials');
        }
      }
    })
    .catch((err) => toast.error(`Signin Failed due to : ${err.message}`));
};


```

Figure 4.23: Signing and fetching data from JSON Server

- **Storing user information in JSON server**

We have simulated of fetching data from API using JSON server. Here in the figure, we can access all of the data by localhost:125. Initially the server is configured as it will run on localhost:3000 port. But I have configured it to run on custom localhost:125 port by changing the start script to "json-server -p 125 --watch db.json", shown in the figure below

```

package.json > ...
1  {
2    "name": "fake-api",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    // Debug
7    "scripts": {
8      "test": "echo \"Error: no test specified\" && exit 1",
9      "start": "json-server -p 125 --watch db.json"
10    },
11    "author": "",
12    "license": "ISC"
13  }


```

Figure 4.24: package.json file configuration of JSON server

- **Accessing user's identity**

After running the JSON server, we can access the user's identity by hitting the URL localhost:125/user. It will give us user identity as an object.

```
"user": {  
    "img": "images/dummy-profile.jpg",  
    "company": "Brain Station 23",  
    "name": "Omar Farook",  
    "role": "Web Developer",  
    "id": 1804125,  
    "description": " Lorem ipsum dolor sit amet consectetur adipisicing elit. Rem rerum non illum, labore architecto delectus.",  
    "mobile": "01867252378",  
    "phone": "02-8620542",  
    "email": "omarfarook666@gmail.com",  
    "address": "146/2 Azimpur, Dhaka"  
},
```

Figure 4.25: Saving user Identity as object format

- **Accessing current projects**

We can also access the user's current running projects by hitting the URL localhost:125/currentProjects. It will give us the user's current project as an array of objects.

```
"currentProjects": [  
    {  
        "projectName": "Social Geek Made",  
        "projectWork": "UI/UX Design",  
        "attach": "5 Attach",  
        "time": "4 months",  
        "icon": "faUiikit",  
        "members": "4 members",  
        "comments": "10 comments",  
        "progress": "60",  
        "timeLeft": "35 days left"  
    },  
    {  
        "projectName": "Practice to Perfect",  
        "projectWork": "Website Design",  
        "attach": "5 Attach",  
        "time": "4 months",  
        "members": "5 members",  
        "comments": "10 comments",  
        "progress": "80",  
        "timeLeft": "15 days left"  
    },  
    {  
        "projectName": "Med UIKit",  
        "projectWork": "Database Design",  
        "attach": "5 Attach",  
        "time": "4 months",  
        "members": "5 members",  
        "comments": "10 comments",  
        "progress": "30",  
        "timeLeft": "1 month left"  
    }  
,
```

Figure 4.26: Saving current projects as array of object format

Chapter 5

Conclusion

In conclusion, joining Brain Station23 for our industrial attachment has been a transformative experience. We immersed ourselves in JavaScript, React.js, and the workings of a service-based software industry. Our project stands as a testament to our newfound skills.

Our React-based Employee Management System showcases the power of modern web technologies to streamline HR processes, improve efficiency, and enhance the overall employee experience. With a user-friendly interface, robust features, and responsive design. We are excited about the possibilities, this system offers to the company, and we look forward to making improvements in this project to ensure growth and innovation in the field of human resource management.