



Internship Report

Najeebullah Sadiq

2129

Innovative Insights and Practical Experience: A Transformative Internship at Zapcom Solutions Private Limited

**Completed by
Najeebullah Sadiq
2129**

For the partial fulfillment of
MCA Degree for Semester III/IV
The discipline of Computer Science and Technology,
Goa Business School,
Goa University
At
Zapcom Solution Ltd.
Gama building, Sigma tech park, Whitefield, Bangalore

Under the guidance of

Mr. Amit Gehelot
(Sr. Software Engineer)

With

Mr. Arvind A
(Sr. Software Engineer)

Rajiv Ram V
(Sr. Software Engineer)



1st June 2023

TO WHOMSOEVER IT MAY CONCERN

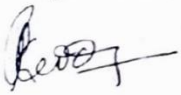
This is to inform you that **Mr. Najeebullah Sadiq**, student of Master of Computer Applications (MCA) of Goa University, Goa, is currently undergoing his final semester project (Semester III/IV) at our company, **Zapcom Solutions Pvt. Ltd** from 4th January, 2023.

During his tenure he has met the expectations of his team lead/mentor/guide and found to be regular and sincere.

This letter is being issued on his request to be submitted with the project report at Goa University.

The final internship completion letter will be provided on completing his internship.

For Zapcom Solutions Pvt. Ltd.


Srinivas Reddy Kothakota
Chief Operating Officer



INDIA
Zapcom Solutions Pvt. Ltd
9th Floor, Gamma Tower, Sigma Soft Tech Park,
Whitefield, Bangalore - 560066
Ph: +91-80-67232300

www.zapcg.com

USA
Zapcom Group Inc.
105 Decker Court, Ste. 810 Irving,
TX 75062.
Ph: (972)441-2081



GOA BUSINESS SCHOOL

CERTIFICATE OF EVALUATION

This is to certify that **Mr. Najeebullah Sadiq** has successfully completed his internship at **Zapcom Solution Ltd at Whitefield Bangalore** in partial fulfillment of the award of the degree in Master of Computer Application.

Examiner 1

Examiner 2

Place: Goa University
Date: 16/06/2023

Dean, Goa Business School

Acknowledgment

My internship at Zapcom Solutions Private Limited has been an incredible opportunity for my personal and professional development. Working with a skilled and diverse team allowed me to expand my knowledge and gain a deeper understanding of the industry. I was entrusted with real-world projects that helped me apply theoretical knowledge in practical settings, enhancing my technical abilities and boosting my confidence.

First and foremost, I would like to extend my heartfelt thanks to the management of Zapcom Solutions Private Limited for providing me with the opportunity to undertake this internship. Their trust, guidance, and support throughout the internship period have been invaluable.

I am immensely grateful to my mentor, **Mr. Amit Gehelot** (Sr. Software Engineer) for their unwavering guidance, mentorship, and valuable insights. His expertise and willingness to share knowledge have been instrumental in shaping my understanding of the industry and enhancing my professional skills.

I am deeply grateful to **Mr. Arvind A** (Sr. Software Engineer) for his exceptional guidance and unwavering support during my internship at Zapcom Solutions.

I am deeply grateful to the entire team at Zapcom Solutions for their extraordinary support and belief in me as an Afghan intern. Despite the challenges faced by others, the company took a chance on me and provided me with this invaluable internship opportunity. I am humbled and appreciative of their inclusive mindset and commitment to diversity.

I am incredibly grateful for the unwavering support and motivation provided by **Ms. Savita Carriappa** (COS, Marketing Head). Her guidance and encouragement have been invaluable to me.

From offering words of support during challenging times to providing motivation to strive for success, she has been an incredible source of inspiration.

I extend my heartfelt gratitude for the exceptional education and guidance provided during my time at Goa University. The commitment to academic excellence and the dedication of the professors have been invaluable.

Thank you for equipping me with knowledge, fostering critical thinking, and creating a supportive learning environment.

I am indebted to my friends and family for their constant encouragement, understanding, and motivation throughout this internship journey. Their unwavering support has been a constant source of strength for me.

Table of Contents

Acknowledgment	5
Table of Contents	6
Introduction	9
Company profile	10
Project-1 PediaSure from Capillary	12
Tools and Technologies.....	12
My Contributions:	12
Project – 2 Zapcom API service	14
Tools and Technologies.....	14
My contributions:.....	14
Screenshots	15
Project- 3 Suntec admin portal	16
Tools and Technologies.....	16
My Contributions:	16
Project-4 Dynamic form	17
Tools and Technologies.....	17
Frontend	17
Backend	17
Screenshots	18
Tools and Technologies used.....	21
JavaScript:	21
CSS:	21

React.JS:	21
Redux:	21
Formik:	22
Yup:	22
Axios:	22
Node.JS	22
Express.JS:	22
Mongoose:	23
Certifications Completed under Internship and Self Study ..	25
Udemy	25
LinkedIn Learning	25
Conclusion and Summary of Internship:	26
Reflections/Experiences of Internship:	27
Internship Timeline	28
January 2023	28
Week 1. 4th to 6th:	28
Week 2. 9th to 13th:	28
Week 3 16th to 20th:	29
Week 4 23th to 27th:	29
February 2023	30
Week 1 30th to 3th:	30
Week 2 6th to 10th:	31
Week 3 13th to 17th:	32
Week 4 20th to 24th:	32
March 2023	33

Week 1 27th to 3rd:	33
Week 2 6th to 10th:	34
Week 3 13th to 17th:	35
Week 4 20th to 24th:	36
Week 5 27th to 31st :	37
April 2023	38
Week 1 3rd to 7th :	38
Week 2 10th to 14th :	39
Week 3 17th to 21st :	39
Week 4 24th to 28th :	39
May 2023	40
Week 1 2nd to 9th :	40
Week 2 8th to 12th:	40
Week 3 5th to 19th:	41
Week 4 22nd to 26th :	41
June 2023	42
References	43

Introduction

Embarking on an internship journey at ZapCom Solutions Private Limited has been a remarkable experience that has allowed me to bridge the gap between academic learning and practical application.

As an aspiring MERN stack professional, this internship has provided me with a platform to immerse myself in real-world projects and gain hands-on experience in a dynamic and innovative environment.

Working alongside a talented team of professionals, I have been able to enhance my skills, expand my knowledge, and develop a deeper understanding of the intricacies of the MERN stack.

This introduction sets the stage for a comprehensive exploration of my internship, capturing the transformative moments, valuable insights, and key learnings that have shaped me into a more proficient and industry-ready professional.

Company profile

Zapcom is a leading product engineering and technology services company specializing in the design and development of custom software solutions. With a focus on the Travel, Hospitality, Fin-tech, and Retail sectors, Zapcom leverages cutting-edge technology to create innovative solutions that drive business growth and enhance customer experiences.

Zapcom offers a wide range of services tailored to meet the unique needs of our clients

1. **Software Development:** Zapcom's expert team of developers and engineers build custom software solutions from the ground up, ensuring seamless integration and optimal performance.
2. **Product Engineering:** Zapcom provide end-to-end product engineering services, including ideation, prototyping, development, and maintenance, to transform ideas into market-ready products.
3. **Mobile Application Development:** With a mobile-first approach, Zapcom creates intuitive and feature-rich mobile applications for iOS and Android platforms, ensuring a seamless user experience.
4. **Web Application Development:** Zapcom team excels in developing scalable and secure web applications that empower businesses to streamline operations, improve efficiency, and enhance customer engagement.
5. **UI/UX Design:** Zapcom prioritizes user-centric design, delivering visually appealing and intuitive user interfaces that provide an exceptional user experience across all platforms.

Zapcom is a trusted provider of custom software solutions for the Travel, Hospitality, Fin-tech, and Retail industries

Offices:

Headquarters:

- Dublin-California, USA

Support offices:

- Bangalore, Hyderabad, India
- Dallas – Texas, USA
- Heredia – Costa Rica

ZapCom in numbers.

30+

Happy Customers

With global presence and 95% client retention.

100+

Projects

Delivered projects of all types and sizes since inception.

400+

Global Builders

Skilled with product mindset, innovation and agility.

Leadership Team.



Kishore Pallamreddy
Executive Chairman & CEO



Sai Konda
Chief Technology Officer



Joan Khuel
Advisor



Srinu Kothakota
Chief Operating Officer



Jim Fitzpatrick
Advisor



Deepak Puranam
Head of Delivery



Our Customers



Project-1 PediaSure from Capillary

Capillary Technologies is an enterprise-grade SaaS technology provider. Capillary's AI-enabled loyalty management platform offers brands comprehensive solutions for customer engagement and experiential loyalty to achieve business goals and deliver a great consumer experience.

The project I contribute is called PediaSure. It is a product code scanner that allows users to scan QR codes on products. Upon scanning, users can redeem rewards associated with the product. The redeemed rewards can be utilized by the user in future transactions. Additionally, the project includes a feature where scanning the QR code of a product displays the gradients or color variations of the product.

The project aims to enhance user engagement, loyalty, and the overall shopping experience.

Tools and Technologies

- React.Js
- Next.js
- Material UI
- Formik
- Yup
- Jest
- Prettier
- Vs code

My Contributions:

1. Developed the admin portal with a user-friendly login interface, ensuring secure access through login form validation and handling.
2. Implemented a dashboard page providing an overview of key metrics and important information for easy monitoring and decision-making.
3. Created functionality for displaying total receipts, enabling administrators to track and analyze overall receipt activity within the system.
4. Designed a dedicated page for pending receipts, allowing administrators to review and process pending receipt submissions efficiently.
5. Developed a separate page for approved receipts, showcasing the list of receipts that have been successfully approved and processed.
6. Implemented a user page, enabling administrators to view and manage user data, including user profiles, permissions, and other relevant details.
7. Created a page specifically for rejected receipts, facilitating easy identification and management of receipts that did not meet approval criteria.
8. Implemented validation mechanisms and input handling across all pages, ensuring data integrity and preventing erroneous or incomplete information from being processed.

By developing the admin portal and its various components, I contributed to creating a comprehensive system for receipt management and user administration. The implementation of

form validation, input handling, and dedicated pages for different receipt statuses streamlined the administrative tasks and improved the overall efficiency and accuracy of the system.

9. Developed the customer side app, featuring intuitive user interfaces for login, signup, forgot password, change password, and edit password functionalities.
10. Implemented form validation to ensure accurate and secure user input, preventing the submission of incomplete or incorrect data.
11. Created a login page that allows customers to securely log into their accounts using their credentials, providing a seamless and secure authentication process.
12. Developed a signup page where new customers can create accounts by providing necessary information, ensuring a smooth onboarding experience.
13. Implemented a forgot password functionality, enabling customers to reset their passwords securely through a password reset page, in case they forget their login credentials.
14. Developed a change password page, allowing customers to update their passwords and enhance the security of their accounts.
15. Created an edit profile page that enables customers to manage their personal information, including updating their addresses, contact details, and other relevant data.
16. Implemented form handling mechanisms across all pages, ensuring smooth data submission and efficient processing of customer requests.
17. Developed Manage Address page allowing customers to update and add their address.

Through my contribution to the customer-side app, I have played a significant role in enhancing the user experience and functionality of the application. The implementation of user-friendly UI pages, along with form validation and handling, ensures a seamless and secure customer journey, from account creation to profile management.

I am not allowed to put screenshots.

Project – 2 Zapcom API service

The project at hand is a comprehensive software solution developed to streamline and enhance various aspects of a specific domain or problem space. It encompasses a range of functionalities and features aimed at addressing specific needs and requirements. Through meticulous planning, development, and implementation, the project aims to improve efficiency, automate processes, and deliver tangible benefits to its users.

Tools and Technologies

- React.Js
- Material UI
- CSS
- Prettier
- Vs code

My contributions:

1. Developed pages for the customer API documentation, including the Getting Started page, Requirement page, Changelog page, and Loading Supply page.
2. Created a user-friendly Getting Started page that provides an overview of the API, including installation instructions, authentication methods, and basic usage examples.
3. Designed a Requirement page that outlines the prerequisites and dependencies for integrating the API, ensuring a smooth and error-free integration process.
4. Developed a Changelog page to keep customers informed about the latest updates, enhancements, and bug fixes in the API, enabling them to stay up-to-date with the changes.
5. Created a Loading Supply page that provides detailed information on how to load and supply data using the API, including the required parameters and formatting guidelines.

Screenshots



HOME

GALAXY API

FRAUD PREVENTION

LODGING SUPPLY

BLOG

SUPPORT

CUSTOMER API

GETTING STARED

REQUIREMENTS

CHANGELOG

Customer API

POST	/api/customer	Creates the specified model.
GET	/api/customer	GetAll Customers Instiutionalcustomers Institutionalemployees
PUT	/api/customer	Updates the specified model.
POST	/api/customer	Gets by list of Ids.
POST	/api/Customer/{id}	get by identifier as an asynchronous operation.
POST	/api/Customer/{externalId}	get by external id as an asynchronous operation.

Project- 3 Suntec admin portal

This app is a powerful and user-friendly solution designed to simplify and streamline various administrative tasks. With key functionalities such as member and merchant management, receipt tracking, campaign and communication management, and content control, this app empowers administrators to efficiently handle essential processes. It includes features for tasks and approval workflows, finance management, CRM workflows, and comprehensive admin management and settings. The app also provides intuitive dashboards and reports, offering valuable insights and facilitating data-driven decision-making. With its seamless functionality and user-centric design, the app enhances productivity and effectiveness in managing administrative operations.

Tools and Technologies

- React.Js
- Material UI
- CSS
- Jest
- Prettier
- Redux
- Vs codes

My Contributions:

1. Developed a home page for the app
2. Developed the routing functionality for the home page sidebar of the app, ensuring smooth navigation and seamless user experience.
3. Implemented a dynamic and responsive sidebar menu that allows users to easily access different sections and features of the app.

This is my first sprint contribution the second sprint will start on the 14th of June

I am not allowed to put screenshots

Project-4 Dynamic form

Dynamic Form is a cutting-edge web application I built on the MERN (MongoDB, Express.js, React, Node.js) stack

At its core, Dynamic Form leverages MongoDB as the backend database, providing a reliable and scalable storage solution for form data. The integration of Mongoose ensures efficient data modeling and interaction with the database.

Express.js powers the backend framework, handling API routes and enabling smooth communication between the frontend and the server. Node.js serves as the runtime environment, ensuring high performance and scalability.

On the frontend, Dynamic Form harnesses the power of React to deliver a dynamic and interactive user interface. The application allows users to dynamically generate and customize forms by adding various field types, such as text inputs, checkboxes, and dropdowns. React components and state management with Redux ensure a smooth and responsive user experience.

Formik and Yup are integrated to facilitate form validation and error handling, ensuring accurate and secure data input. Axios is used for efficient API communication, enabling seamless interaction between the frontend and backend.

Tools and Technologies

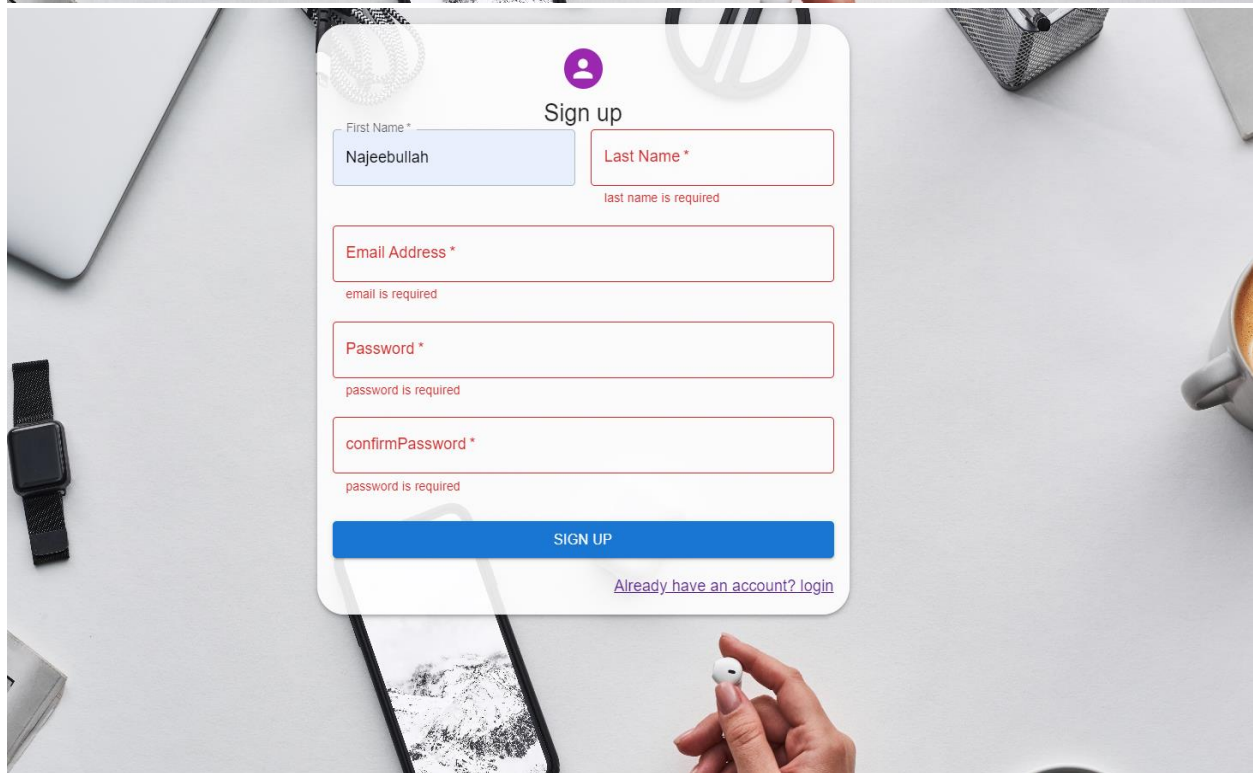
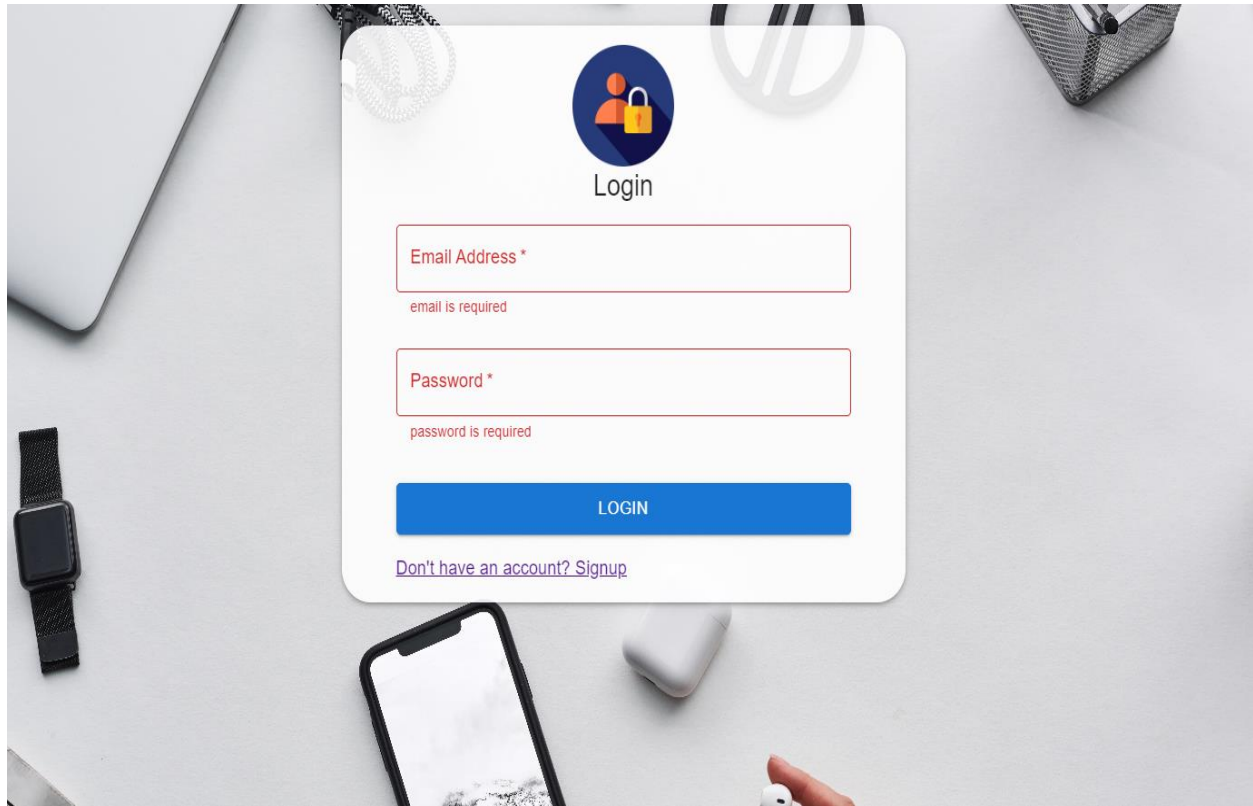
Frontend

- React.Js
- Material UI
- CSS
- Formik
- Yup
- Prettier
- Redux
- Axios
- Vs code

Backend

- Node.Js
- Express.Js
- MongooseMongoDBb
- Postman

Screenshots



1 Personal Information 2 Family information 3 Friend Information 4 Review Form



Favourite Book
The way of life

Favourite Sport
football

Permanent Address
The way of life

Enter college
Banglore

Blood Group
O+ve

NEXT

LOGOUT

1 Personal Information 2 Family information 3 Friend Information 4 Review Form



Father Name
Mohmmad Zahir

Brother Name
Ahmad

Email Address
Khan@gmail.com

Address
Hydrabad

Country Name
Afghanistan

BACK NEXT

LOGOUT

Progress: 1 Personal Information, 2 Family Information, 3 Friend Information, 4 Review Form

FriendName Name: Khalid

Address: Bangalore

FrienEmail Address: Khalid@gmail.com

Enter gender: Female

Buttons: BACK, NEXT, LOGOUT

Progress: 1 Personal Information, 2 Family Information, 3 Friend Information, 4 Review Form

Personal Information

book	The way of life
sport	football
address	The way of life
college	Bangalore
BloodGroup	O+ve

Family Information

FatherName	Mohammad Zahir
BrotherName	Ahmad
email	Khan@gmail.com
Address	Hydrabad
Country	Afghanistan

Friend Information

FriendName	Khalid
Address	Bangalore
FrienEmail	Khalid@gmail.com
gender	Female

Buttons: BACK, SUBMIT

Tools and Technologies used

JavaScript:

JavaScript is a high-level programming language that is primarily used to add interactivity and dynamic features to websites. It is a versatile language that can be executed on both the client side (in a web browser) and the server side (using Node.js).

CSS:

CSS (Cascading Style Sheets) is a style sheet language used to describe the presentation and formatting of a document written in HTML or XML. It provides a set of rules and properties that define how elements in a web page should be displayed, such as their colors, sizes, positions, and animations. CSS allows web developers to separate the design and layout of a website from its content, making it easier to maintain and update the appearance of multiple web pages simultaneously. By applying CSS styles, developers can create visually appealing, consistent, and responsive web pages that adapt to different screen sizes and devices.

React.JS:

React.JS is a JavaScript library for building user interfaces that offers a component-based approach, allowing developers to create reusable and self-contained UI components. It utilizes a virtual DOM to efficiently update and render components, resulting in optimized performance. With JSX syntax, developers can write HTML-like code within JavaScript, simplifying the process of building UI elements. React follows a unidirectional data flow, ensuring predictability and facilitating debugging. The introduction of React Hooks provides a convenient way to use state and other React features in functional components. Overall, React.JS offers a powerful and efficient solution for building dynamic and interactive user interfaces in web applications.

Redux:

Redux is a centralized state management library for JavaScript applications. It uses a single store to hold the application state and dispatches actions to modify the state. Reducers are pure functions that handle state updates based on dispatched actions. With Redux, developers can easily manage and share application state across components, ensuring predictable data flow and simplifying debugging.

Formik:

Formik is a lightweight library for managing forms in React applications. It simplifies form handling by providing features for managing form state, validation, and submission. With Formik, developers can easily set up forms, handle input changes, and handle form submission with minimal boilerplate code. It offers seamless integration with React components and supports advanced features like asynchronous form handling. Formik streamlines the process of building and managing forms, making it a popular choice for form management in React applications.

Yup:

Yup is a JavaScript schema validation library commonly used with Formik to validate form inputs in React applications. It provides a simple and declarative way to define validation rules for form fields. With Yup, developers can easily set up validation schemas, specify rules for each field, and validate input values. Yup supports various types of validations, including required fields, string lengths, email formats, and more. It integrates seamlessly with Formik, enabling efficient and robust form validation. Yup simplifies the process of validating form data, ensuring that inputs meet the specified requirements before submission.

Axios:

Axios is a JavaScript library for making HTTP requests. It offers a simple and intuitive API for sending requests, handling responses, and managing various aspects of the HTTP communication process. With Axios, developers can easily interact with RESTful APIs, handle authentication, and utilize advanced features like request cancellation and interceptors. Its promise-based approach simplifies handling asynchronous operations, making Axios a popular choice for HTTP requests in both browsers and Node.js environments.

Node.JS

Node.js is a server-side JavaScript runtime environment that enables developers to build fast and scalable web applications. It utilizes an event-driven, non-blocking I/O model and has a vast ecosystem of libraries and modules available through npm. With Node.js, developers can handle network requests, access databases, and execute server-side logic, making it a popular choice for building APIs and real-time applications.

Express.JS:

Express.js is a minimal and flexible web application framework for Node.js. It simplifies web development by providing a robust set of features for routing, handling requests, and integrating with databases. With Express.js, developers can quickly build scalable and efficient web applications and APIs.

Mongoose:

Mongoose is a JavaScript library for MongoDB that simplifies database interaction in Node.js applications. It provides an intuitive way to define schemas, create documents, perform queries, and establish relationships. With Mongoose, developers can efficiently work with MongoDB using expressive and convenient APIs.

Mongodb:

MongoDB is a scalable and flexible NoSQL database that stores data in JSON-like documents. It supports dynamic querying, high availability, and horizontal scaling, making it suitable for diverse applications and large datasets.

Postman:

Postman is an essential API development tool that provides a user-friendly interface for sending, testing, and documenting HTTP requests and APIs. It streamlines API development workflows with features like request management, automated testing, and interactive documentation.

VS Code:

VS Code is a lightweight and customizable source code editor with robust features for coding, debugging, and version control. It supports a wide range of programming languages and frameworks, and its vibrant ecosystem of extensions enhances its functionality to suit different developer needs.

Git:

Git is a distributed version control system that enables developers to track and manage code changes efficiently. It simplifies collaboration, allows for branching and merging, and provides a reliable history of project revisions. Git is a fundamental tool for version control in software development.

GitLab:

GitLab is a web-based platform that provides a complete DevOps lifecycle solution, offering robust Git repository management, collaboration features, and tools for project management, CI/CD, code review, and issue tracking. It simplifies the software development process and supports end-to-end development and collaboration.

AWS:

AWS (Amazon Web Services) is a leading cloud computing platform that offers a wide range of services for building scalable and flexible applications. It provides virtual servers, storage, databases, machine learning, and more, along with tools for monitoring and deployment. AWS enables developers to leverage cloud infrastructure and services to accelerate application development and achieve high scalability and reliability.

Azure:

Azure is a robust cloud computing platform offered by Microsoft. It provides a wide range of services and tools for building, deploying, and managing applications in the cloud. With Azure, developers can create virtual machines, store data, build web and mobile apps, implement AI, and utilize serverless computing. It offers global scalability and integration with Microsoft's ecosystem.

Jira:

Jira is a versatile project management and issue-tracking tool that enables teams to efficiently plan, track, and collaborate on projects. With customizable workflows, agile features, and seamless integrations, Jira simplifies project management and enhances team productivity.

Certifications Completed under Internship and Self Study

Udemy

1. JavaScript - The complete guide (Beginner + Advanced)
2. React – The complete Guide (incl Hooks, React Router, Redux)
3. Node.js, Express, MongoDB & More
4. Testing React with Jest and React Testing Library (RTL)

LinkedIn Learning

1. Agile Foundation Course
2. Scrum
3. Node.js
4. Express.js

Conclusion and Summary of Internship:

During my internship at ZapCom Solutions Private Limited, I had the opportunity to work on several significant projects that allowed me to contribute to the development of key functionalities in the admin portal and customer-side app.

In the admin portal, I played a crucial role in developing the login functionality, including UI form validation and secure access. I also implemented various features such as the dashboard, total receipts management, pending receipts management, approved receipts management, user management, and rejected receipts page. Additionally, I focused on ensuring validation and proper handling of inputs throughout these pages, enhancing data accuracy and user-friendliness.

For the customer-side app, I took charge of developing essential UI pages such as login, signup, forgot password, change password, edit password, and manage address. I incorporated form validation to ensure accurate and secure data input from customers and successfully handled form submissions, integrating them with the backend database. These efforts contributed to a smooth user experience and enhanced functionality.

Throughout my internship, I had the opportunity to work with technologies such as React, Formik, Yup, Redux, and Axios, gaining valuable hands-on experience in frontend development and form handling. Collaborating with a talented team, I received guidance and support, expanding my technical skills and knowledge in software development.

Overall, my internship at ZapCom Solutions Private Limited was a transformative experience that allowed me to apply my knowledge and skills in a professional setting. Through my contributions to the admin portal and customer-side app, I developed a deeper understanding of UI development, form handling, and data management. I am grateful for the learning opportunities, the support of my colleagues, and the chance to contribute to impactful projects. This internship has undoubtedly prepared me for future endeavors in the field of software development.

Reflections/Experiences of Internship:

My internship at ZapCom Solutions Private Limited was a truly enriching experience that provided me with valuable insights and learnings in the field of software development. Throughout my internship, I encountered various challenges and gained valuable experiences that have shaped me both personally and professionally.

First and foremost, the opportunity to work on real-world projects was invaluable. Being part of the development team allowed me to apply my theoretical knowledge to practical scenarios, understand the development lifecycle, and gain hands-on experience with industry-standard tools and technologies.

Collaborating with experienced professionals was another highlight of my internship. Working alongside talented developers and mentors provided me with guidance, support, and valuable feedback. Their expertise and willingness to share their knowledge enhanced my technical skills and helped me improve as a developer.

The internship also exposed me to the importance of effective communication and teamwork. Collaborating with team members, understanding project requirements, and coordinating tasks taught me the significance of clear and concise communication for successful project execution.

One of the most significant lessons I learned during my internship was the importance of adaptability and embracing new technologies. As projects evolved and requirements changed, I had to quickly adapt and learn new tools and frameworks to meet the project's needs. This experience strengthened my problem-solving skills and taught me the value of being flexible in a dynamic work environment.

Furthermore, the internship provided me with insights into the professional work environment and corporate culture. I learned about the importance of meeting deadlines, managing time efficiently, and maintaining a high level of professionalism in all aspects of my work.

Overall, my internship at ZapCom Solutions Private Limited was a transformative experience that allowed me to grow both technically and personally. The practical exposure, guidance from experienced professionals, and the opportunity to work on real-world projects have equipped me with valuable skills and knowledge that will undoubtedly benefit me in my future career. I am grateful for the opportunity and look forward to leveraging these experiences to excel in the software development industry.

Internship Timeline

Date of Joining: 4th January 2023

January 2023

Week 1. 4th to 6th:

- Setup and installed essential tools such as VS Code, Node.js, and MongoDB to create a conducive development environment.
- Participated in meetings with the IT team and HR team to discuss internship objectives, and project assignments, and finalize the technologies I would be learning and working on during my internship.
- Installed necessary tools and dependencies for the MERN (MongoDB, Express.js, React, and Node.js) stack, enabling me to work on projects utilizing these technologies.
- Presented my university projects to Mr. Amit Ghelot, discussing their scope, challenges, and potential areas of improvement.
- Collaborated with Mr. Amit Ghelot which assigned my Mentor to determine which courses would best complement my internship experience, ensuring a well-rounded and comprehensive learning journey.

Week 2. 9th to 13th:

- JavaScript Fundamentals:
- Explored the concepts of functions in JavaScript, including working with function objects, arrow functions, anonymous functions, and utilizing default arguments and the rest operator.
- Developed an understanding of creating functions inside functions, enabling modular and reusable code.
- React Components:
- Delved into React components, gaining proficiency in JSX syntax and understanding the process of passing data through props.
- Collaborated with the CEO, discussing project requirements and aligning goals for successful implementation.
- DOM Manipulation:
- Acquired hands-on experience working with the Document Object Model (DOM), evaluating and manipulating DOM elements.
- Explored concepts such as changing DOM properties, traversing the DOM, and dynamically creating elements using JavaScript.
- React Component Development:
- Distinguished between functional and class components, comprehending their respective use cases and advantages.
- Gained proficiency in exporting and importing models within React projects.

- React lifecycle methods, managing component states, and effectively working with events.
- Embraced React Hooks, particularly the useState hook, to simplify state management within functional components.

Week 3 16th to 20th:

- JavaScript Data Structures:
- Explored different ways of creating arrays and gained familiarity with important array methods such as concat(), indexOf(), find(), findIndex(), forEach(), includes(), push(), pop(), unshift(), shift(), sort(), splice(), and slice().
- Delved into array destructuring, allowing for efficient manipulation of array elements.
- Maps, Sets, and Objects in JavaScript:
- Studied maps and their methods, sets and their methods, and implemented map functions using arrow functions and this keyword.
- Worked with objects, including adding and removing object properties, utilizing the Object.assign() method, and practicing object destructuring.
- Explored getter and setter methods for objects.
- React Data Flow:
- Learned about two-way binding in React, enabling data synchronization between parent and child components.
- Explored techniques for passing data from parent to child components and vice versa.
- Implemented rendering lists and conditional content in React components.
- Styling React Components:
- Explored various approaches to styling React components, including conditional and dynamic styling.
- Gained experience with popular styling methodologies, such as using styled components and understanding CSS models.
- Debugging and Development Tools:
- Developed skills in understanding React error messages and analyzing code flow and warnings.
- Utilized breakpoints and leveraged React DevTools for efficient debugging and code analysis.

Week 4 23th to 27th:

- Object-Oriented Programming (OOP) and Classes in JavaScript:
- Explored constructor functions and their differences compared to classes.
- Deepened my understanding of prototypes, working with prototypes, and utilizing the prototype chain.
- Examined the global object and its significance.
- Practiced implementing classes and prototypes, understanding their roles in object-oriented programming.

- Learned about methods in classes and constructors, utilizing built-in prototypes, and setting/getting prototypes.
- Object-Oriented Programming Practice:
- Engaged in practical exercises focused on object-oriented programming principles and class implementation.
- Explored passing method references and manipulating DOM elements.
- Gained familiarity with inheritance and its application in JavaScript.
- Advanced React Concepts:
- Explored advanced React features such as working with fragments, portals, and refs.
- Differentiated between controlled and uncontrolled components, understanding their respective use cases.
- Delved into handling side effects, utilizing reducers, and leveraging the `useEffect()` hook.
- Practiced adding nested properties as dependencies to the `useEffect()` hook for effective dependency tracking.

February 2023

Week 1 30th to 3th:

- DOM and Browser APIs:
- Worked with the DOM to access and manipulate elements on web pages.
- Learned about getting element box dimensions, element sizes, and positions.
- Explored prototypes in the DOM and utilized them for dynamic script loading, setting timers and intervals, and working with location, history, and navigator objects.
- Gained familiarity with working with dates and understanding their representations in JavaScript.
- Event Handling in JavaScript:
- Practiced event handling in JavaScript, including listening to events and removing event listeners.
- Utilized the event object and learned about event propagation, capturing, and bubbling.
- Explored event delegation, triggering DOM elements programmatically, and working with event handler functions.
- Studied drag and drop theory and its implementation in JavaScript.
- React Context API:
- Utilized the React Context API and the `useContext` hook for managing shared data and making it accessible across components.
- Explored making context dynamic and building and using a custom Context Provider component.
- Gained an understanding of the limitations of the context API and followed the rules of hooks for proper usage.
- React Component Optimization:
- Explored techniques for optimizing React components, focusing on component updates and preventing unnecessary re-evaluation.

- Utilized React.memo, useCallback, and proper state management to minimize re-renders and improve performance.
- Explored the differences between class-based and functional-based components, understanding their respective features and advantages.
- Gained familiarity with lifecycle methods and component context within class-based components.

Week 2 6th to 10th:

- In this week I focused on advanced concepts in JavaScript and React development. Here are the key areas I explored:
- Advanced Function Concepts:
- Studied pure functions and side effects, understanding the difference between impure and pure functions.
- Explored factory functions, closures, memory management, IIFEs (Immediately Invoked Function Expressions), and recursion.
- Asynchronous JavaScript:
- Delved into the execution of synchronous and asynchronous code, understanding the event loop and blocking code.
- Worked with multiple callbacks and setTimeout, and learned about promises, their states, and how to handle errors with async/await.
- Explored advanced promise concepts such as Promise.all() and Promise.race().
- HTTP Requests and API Integration:
- Utilized the Star Wars API to send GET requests using async/await.
- Handled loading and HTTP errors in the request process.
- Explored the use of useEffect for making API requests and sending POST requests.
- Custom React Hooks:
- Built custom React hooks by creating functions that encapsulate reusable logic.
- Configured custom hooks for specific use cases.
- Developed a custom HTTP hook for handling HTTP requests.
- Working with Forms and User Input:
- Explored form submission and retrieving user inputs.
- Added validation to form inputs and handled the "touched" state.
- Implemented functionality to react to lost focus and derive states from user input.

Week 3 13th to 17th:

- Working with HTTP Requests:
- Sent HTTP requests and learned about handling JSON data and parsing responses.
- Promised HTTP requests and explored concepts like sending POST and DELETE requests.
- Implemented error handling and used the Fetch API for making HTTP requests.
- Added custom headers to the requests for authentication and other purposes.
- Introduction to JavaScript Libraries:
- Explored popular JavaScript libraries such as JQuery and Axios.
- Learned about their features and advantages in web development.
- Redux State Management:
- Delved into Redux, a popular state management library.
- Created a Redux store and explored the core principles of Redux.
- Integrated Redux with React components and dispatched actions from within components.
- Worked with multiple states and attached payloads to actions for manipulating data.
- Introduction to Redux Toolkit:
- Explored Redux Toolkit, a comprehensive package for efficient Redux development.
- Added state slices using Redux Toolkit to organize the Redux store.
- Connected Redux Toolkit state with React components and read/dispatched actions from the new slices.

Week 4 20th to 24th:

- Modular JavaScript:
- Explored the concept of modular JavaScript and learned about default exports and dynamic imports.
- Implemented code-splitting techniques to optimize the loading of modules.
- Gained a deeper understanding of module execution and scoping in JavaScript.
- JavaScript Tooling and Workflow:
- Worked with npm packages for managing dependencies and explored their usage in JavaScript projects.
- Configured ESLint for code linting and adhering to coding standards.
- Advanced Redux:
- Explored the integration of Redux with side effects and async code.
- Utilized useEffect hook in combination with Redux for handling asynchronous actions and fetching data.
- Implemented error handling and feedback mechanisms using Redux.
- Utilized Redux Thunk middleware for managing asynchronous actions and integrating with fetch requests.
- Explored the Redux DevTools extension for debugging and monitoring Redux state changes.
- Building a Multi-Page Single Page Application (SPA) with React Router:

- Implemented routing in a React application using React Router.
- Defined routes, including nested routes, and learned about dynamic routing.
- Explored relative and absolute paths for navigation within the application.

March 2023

Week 1 27th to 3rd:

- JavaScript and Browser Support:
- Explored the concept of feature detection and implemented fallback code for better browser compatibility.
- Learned how to use polyfills to provide missing functionalities in older browsers.
- transpile code using tools like Babel to ensure compatibility with a wider range of browsers.
- Develop Share My Place App:
- Implemented index routes and handled data fetching with a loader to enhance the user experience.
- Utilized custom error handling techniques for error management in the application.
- Leveraged the JSON() utility function for processing and manipulating data.
- Implemented dynamic routes and loaders to handle varying data requirements based on user interactions.
- Utilized the useRouteLoaderData() hook for efficient handling of actions and data fetching.
- Develop Food Order App:
- Built upon the concepts and skills acquired in previous projects to develop a food ordering application.
- Implemented features such as user authentication, menu display, cart management, and order placement.
- Leveraged the knowledge of JavaScript and web development concepts to create a seamless and interactive user experience.

Week 2 6th to 10th:

- Formik:
- Implemented Formik library in project for efficient form management.
- Utilized useFormik hook to handle form values, submission, and validation.
- Worked with initial values, form field changes, and form submission using HandleChange, HandleSubmit, and Values.
- Incorporated validation schema using yup for form validation, handling errors, and touched fields.
- Successfully implemented form functionality in the project.
- Async Code:
- Explored different approaches for handling asynchronous code, including callbacks, promises, and async/await.
- Worked on projects that involved asynchronous operations, such as fetching data from APIs and updating the UI accordingly.
- Gained experience in handling asynchronous operations effectively and ensuring proper error handling.
- Adding Authentication to React App:
- Learned about authentication mechanisms in React applications.
- Implemented user authentication features, including login, registration, and logout functionality.
- Ensured authentication persistence to keep users logged in across sessions.
- Implemented auto-logout functionality for enhanced security.
- TypeScript Basics and Integration with React:
- Gained familiarity with TypeScript basics, such as working with arrays, objects, unions, and functions.
- Combined React with TypeScript to leverage static typing and enhanced code readability.
- Utilized bcryptjs library for password hashing and encryption.
- Successfully worked on projects that integrated TypeScript into the development workflow.

Week 3 13th to 17th:

- Working with JavaScript Libraries:
- Explored popular JavaScript libraries like Math, Lodash, JQuery, and Axios.
- Utilized their functionalities to perform complex mathematical calculations, handle data manipulation, make AJAX requests, and interact with the DOM.
- Learned how to import and export modules to organize code and leverage the functionalities provided by these libraries.
- Introduction to Node.js and NPM:
- Learned the basics of Node.js and its role in backend development.
- Explored NPM (Node Package Manager) and its importance in managing dependencies.
- Understood the concepts of asynchronous and synchronous code execution in Node.js.
- Utilized core modules like fs (file system) for reading and writing files, and HTTP for building a simple API.
- Package.json and Dependency Management:
- Learned how to manage project dependencies using the package.json file.
- Differentiated between regular dependencies and development dependencies.
- Explored versioning and updating packages to ensure compatibility and security in the project.
- Event-Driven Architecture and Streams:
- Understood the event-driven architecture of Node.js and its benefits.
- Worked with various types of streams like readable, writable, duplex, and transform streams for efficient data processing.
- Explored how requiring modules work and utilized core modules, developer modules, and third-party modules from npm.
- Introduction to Express and RESTful API Design:
- Explored the Express framework for building web applications and APIs.
- Understood the principles of RESTful API design.
- Implemented HTTP GET and POST request handling using Express.
- Practiced these concepts using tools like Postman for testing and interacting with the APIs.

Week 4 20th to 24th:

- Responding to URL Parameters and Handling Requests:
- Learned how to handle URL parameters in Express and respond accordingly.
- Implemented PATCH and DELETE requests to update and delete resources.
- Explored the concept of middleware and its role in the request-response cycle.
- Worked with third-party middleware like Morgan for logging HTTP requests.
- Utilized params middleware for extracting and validating parameters from incoming requests.
- Implemented serving static files in Express for serving images, CSS, and other static assets.
- Introduction to MongoDB and CRUD Operations:
- Explored the NoSQL database MongoDB and its advantages for handling unstructured data.
- Gained proficiency in performing CRUD operations (Create, Read, Update, and Delete) with MongoDB.
- Connected MongoDB with Express and utilized MongoDB Compass and Mongo Shell for managing and interacting with the database.
- Leveraged Mongoose, an Object Data Modeling (ODM) library, to simplify database operations.
- Created models using Mongoose to define the structure and behavior of MongoDB documents.
- Document Creation, Deletion, and Updates:
- Practiced creating, deleting, and updating documents in MongoDB using Mongoose.
- Implemented validation and data manipulation operations on the documents.
- Learned about the Model-View-Controller (MVC) architecture and its benefits in organizing code and separating concerns.
-
- Importing Development Data and Advanced API Features:
- Imported development data using scripts to populate the database with initial data.
- Implemented API filtering, allowing users to filter resources based on specific criteria.
- Explored advanced filtering options, such as sorting, limiting fields, and pagination, to enhance the functionality of the APIs.

Week 5 27th to 31st :

- Aliasing, Aggregation Pipeline, and Grouping:
 - Explored aliasing in MongoDB to provide alternative names for fields in query results.
 - Learned about the aggregation pipeline, which allows for advanced data manipulations and transformations.
 - Utilized grouping techniques to group and summarize data based on specific criteria.
- Unwinding, Projecting, and Virtual Properties:
 - Utilized the unwinding operation to create multiple documents from a single document array.
 - Worked with virtual properties to define additional properties in MongoDB documents that are derived from existing fields.
- Document Middleware and Hooks in Mongoose:
 - Explored document middleware in Mongoose to add custom behavior before or after specific document operations.
 - Implemented pre and post-hooks in Mongoose to perform actions such as data manipulation or logging before or after certain operations.
- Querying, Aggregating, and Validating Data in Mongoose:
 - Leveraged query and aggregate features in Mongoose to retrieve and manipulate data from MongoDB.
 - Utilized model middleware to add custom logic before or after specific model operations.
 - Explored the built-in data validation capabilities in Mongoose and utilized validators to enforce data integrity.
 - Implemented custom validation in Mongoose to validate data based on specific business rules.
- Debugging and Error Handling in Node.js and Express:
 - Used the `ndb` debugger to debug Node.js applications and track down issues in the code.
 - Handled unhandled routes and implemented a global error handler to catch and handle errors throughout the application.
 - Differentiated between errors occurring in production and development environments and implemented appropriate error-handling strategies.
 - Addressed database errors and handled Mongoose validation errors gracefully.
 - Handled errors outside the scope of Express.js to ensure a robust and error-free application.
- Authentication and User Management:
 - Explored the process of user authentication and authorization.
 - Modeled user entities and implemented user registration and login functionalities.
 - Managed password security and implemented appropriate encryption and hashing techniques.
 - Utilized JSON Web Tokens (JWT) to facilitate secure authentication and session management.
 - Protected routes to ensure that only authenticated users can access certain resources

April 2023

Week 1 3rd to 7th :

- User Authentication and Login:
- Implemented user login functionality, allowing users to authenticate themselves using their credentials.
- Utilized bcrypt to securely compare passwords and ensure authentication accuracy.
- Leveraged Mongoose middleware to handle password hashing and verification seamlessly.
- Advanced Postman Setup:
- Explored advanced features of Postman to enhance API testing and development.
- Set up global variables in Postman for efficient and streamlined testing across different endpoints and requests.
- User Authorization, Roles, and Permissions:
- Implemented user roles and permissions to distinguish between admin and normal users.
- Defined specific authorization rules and access levels based on user roles.
- Password Reset Functionality:
- Implemented a password reset mechanism that allows users to reset their passwords.
- Generated reset tokens and integrated nodemailer to send password reset emails to users.
- Setting New Password and Updating User Profile:
- Developed functionality to allow users to set a new password after a reset.
- Implemented the ability for users to update their profile information and manage their account settings.
- Integration with Personal Project:
- Applied the acquired knowledge and skills to implement login and signup functionality in my personal project.
- Ensured seamless integration between the backend and frontend components of the project.

Week 2 10th to 14th :

- Worked on the DynamicForm project as a learning project to implement the concepts of the MERN (MongoDB, Express.js, React, Node.js) stack.
- Contributed to the IP project, specifically focusing on the development of wiki pages for zapcom.
- Implemented the logout functionality in the DynamicForm project, allowing users to securely log out of the application.
- Developed backend API endpoints for submitting, updating, and retrieving data from the database based on user interactions.
- Played a role in creating and enhancing the documentation pages for the zapcom wiki pages project.
- Researched and organized information about the APIs, documenting their functionalities, usage, and parameters.
- Gained practical experience with the MERN stack, including authentication, data management, and documentation.

Week 3 17th to 21st :

- Participated in the IP project focused on developing wiki pages for Zapcom, contributing to the frontend development.
- Implemented the frontend functionality for updating, submitting, and retrieving data in the DynamicForm project.
- Collaborated with the team to design and implement user-friendly forms and interfaces for data management.
- Utilized React components, state management, and API integration to handle data operations in the frontend.
- Worked on backend to ensure seamless communication and data flow between the frontend and backend systems.
- Gained practical knowledge in frontend development using React and integrating it with backend APIs.

Week 4 24th to 28th :

- Engaged in learning Next.js as a prerequisite for the Capillary project.
- Explored the features and capabilities of Next.js, a popular React framework for server-side rendering and static site generation.
- Studied the Next.js documentation and tutorials to gain a solid understanding of its concepts and best practices.
- Practiced building small Next.js applications to familiarize myself with its file structure, routing system, and data fetching capabilities.

- Implemented dynamic routes, API routes, and server-side rendering in Next.js projects to leverage its powerful features.
- Collaborated with peers and mentors to clarify doubts, share insights, and discuss the application of Next.js in real-world scenarios.
- Developed a foundational knowledge of Next.js, enabling me to confidently contribute to the upcoming Capillary project.

May 2023

Week 1 2nd to 9th :

- Contributed to the migration project from Angular to React in Capillary.
- Implemented the login and sign-in module in React, replacing the existing functionality in Angular.
- Transferred the business logic and UI components from Angular to React, ensuring a seamless transition for users.
- Developed React components such as SLHeader, AddChild, TermAndCondition, and more, as part of the login and sign-in module.
- Ensured the functionality and behavior of the login and sign-in module were consistent with the original implementation.
- Collaborated with the team to address any challenges or issues during the migration process and worked towards effective solutions.
- Conducted thorough testing and debugging to verify the correctness and stability of the migrated login and sign-in module.
- Successfully completed the migration of the login and sign-in module, contributing to the overall progress of the React migration project at Capillary

Week 2 8th to 12th:

- Contributed to the migration project from Angular to React in Capillary.
- Developed the UI components for the account and admin modules in React, replacing the existing Angular implementation.
- Ensured the UI design and layout were consistent with the original Angular version, maintaining a seamless user experience.
- Implemented functionality such as user account management, admin dashboard, and other features required for the account and admin modules.
- Collaborated with the team to understand the requirements and specifications of the account and admin modules.

- Worked closely with designers and backend developers to integrate the UI components with the corresponding backend APIs.
- Conducted thorough testing and debugging to ensure the UI components functioned correctly and met the project's quality standards.
- Participated in code reviews and actively contributed to the improvement of the React migration project.
- Successfully completed the development of the UI components for the account and admin modules, making significant progress in the Angular to React migration at Capillary.

Week 3 5th to 19th:

- Reviewed all the comments and feedback provided by the team for the Capillary Angular to React migration project.
- Implemented the SuccessMessagePage component, which displays success messages to users in a visually appealing and informative way.
- Configured testing in the project, ensuring that all components and functionality are thoroughly tested for quality assurance.
- Conducted testing specifically for the static pages of the Login-Signup module, ensuring they render correctly and function as intended.
- Configured TypeScript in the project, enabling type checking and improved code maintainability.
- Updated the styles of the admin, login-signup, and account modules to match the theme from Material UI, providing a cohesive and visually pleasing user interface.
- Collaborated with designers and frontend developers to ensure the updated styles aligned with the overall design guidelines and branding of Capillary.
- Conducted thorough testing and debugging to ensure the updated styles did not introduce any visual or functional issues.
- Actively participated in code reviews and addressed any issues or suggestions related to the implementation of the SuccessMessagePage, testing configuration, TypeScript integration, and style updates.
- Successfully completed the tasks, contributing to the overall enhancement and quality improvement of the Capillary Angular to React migration project.

Week 4 22nd to 26th :

- Implemented form validation for the account, admin, and login-signup modules as part of the Angular to React migration project.
- Applied validation rules and logic to ensure that user input in the forms meets the required criteria and constraints.
- Handled validation errors and provided appropriate error messages to guide users in correcting their input.
- Utilized form validation libraries or built custom validation functions to validate fields such as email, password, username, and other relevant form inputs.

- Collaborated with the team to define the validation requirements and ensure consistency across the different modules.
- Conducted thorough testing to validate the effectiveness of the implemented form validation and to identify and fix any potential issues.
- Made necessary adjustments and improvements based on feedback and review from the team.
- Documented the implemented form validation logic and guidelines for future reference and maintenance.
- Contributed to the overall success of the Angular to React migration by ensuring the functionality and integrity of the forms through proper validation.

June 2023

Week 1 29th to 2nd :

- Implemented the Home page for the Suntec project, which serves as the main landing page for the application.
- Created and structured the necessary components for the Home page, including the header, sections, and other relevant elements.
- Implemented the routing setup for the Suntec project, ensuring that the Home page is properly linked and accessible within the application.
- Utilized appropriate React Router functionalities to handle navigation and rendering of the Home page.
- Ensured responsive design and compatibility across different devices and screen sizes for the Home page.
- Collaborated with the team to gather requirements, discuss design decisions, and align the implementation with project objectives.
- Conducted testing and debugging to verify the functionality and performance of the Home page, addressing any issues or bugs encountered.
- Documented the implementation details and any important considerations for the Home page, providing reference for future maintenance and enhancements.
- Contributed to the overall success of the Suntec project by delivering a fully functional and visually appealing Home page with proper routing and component structure.

References

1. <https://www.udemy.com/>
2. <https://www.linkedin.com/learning/>
3. <https://www.w3schools.com/>
4. <https://docs.github.com/en>
5. <https://react.dev/learn>
6. <https://redux.js.org/>
7. <https://axios-http.com/docs/intro>
8. <https://formik.org/docs/overview>
9. <https://mui.com/material-ui/getting-started/overview/>
10. <https://reactrouter.com/en/main>
11. <https://nodejs.org/en/docs>
12. <https://www.npmjs.com/package/dotenv>
13. <https://expressjs.com/>
14. <https://pypi.org/project/bcrypt/>
15. <https://www.mongodb.com/docs/>
16. <https://mongoosejs.com/docs/>
17. <https://jwt.io/introduction>
18. <https://learning.postman.com/docs/publishing-your-api/documenting-your-api/>
19. <https://www.postman.com/api-documentation-tool/>
20. https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API