Internship Report

An Internship Report for

Course code and Course Title: CSA-652 Industry Internship

Credits: 16

Submitted in partial fulfillment of Masters Degree

in MCA

by

AIWINRAJ KURUPPATH SREEJAYAN

Seat Number : 2267

ABC ID : 884578755110

PRN : 202200128

Under the Mentorship of

ANKITA GAONKAR

Goa Business School
Discipline of Computer Science & Technology

GOA UNIVERSITY

Date: April 2024

Examined by:

Seal of the School/Dept

DECLARATION BY STUDENT

I hereby declare that the data presented in this Internship report entitled, **"Report of Internship at Eejak Technologies Private Ltd"** is based on the results of investigations carried out by me in the Discipline of computer science & Technology at the Goa business school, Goa University, under the mentorship of **Ms. Ankita Gaonkar** and the same has not been submitted elsewhere for the award of a degree or diploma by me. Further, I understand that Goa University or its authorities/College will not be responsible for the correctness of observations / experimental or other findings given the internship report/work. I hereby authorize the University/college authorities to upload this dissertation on the dissertation repository or anywhere else as the UGC regulations demand and make it available to any one as needed.

Signature and Name of Student
Seat no: 2267

Date: 27/04/2024
Place: Goa University

COMPLETION CERTIFICATE

This is to certify that the internship report **"Report of Internship at Eejak Technologies Private Ltd"** is a bonafide work carried out by **Mr Aiwinraj Kuruppath Sreejayan** under my mentorship in partial fulfillment of the requirements for the award of the degree of MCA in the Discipline computer science & technology at the Goa business school, Goa University.

Signature and Name of Mentor

Date :

Signature of Dean of School/HoD

School/Department Stamp

Date:

Place: Goa University

CONTENTS

INTERNSHIP CERTIFICATE

Eejak
Empowering Your Dreams

Date: 5th June, 2024

TO WHOM IT MAY CONCERN

This is to certify that **Mr Aiwinraj K.S** student of **Goa University** has successfully completed **10 weeks internship training.**

Mr Aiwinraj joined our organization M/s EEJAK TECHNOLOGIES PRIVATE LIMITED at Goa branch office #305, Gera Imperium Grand, Patto, Panjim as an intern for frontend reactjs development project from dated: 18th March 2024 to 30th May 2024. During his internship he has worked in reactjs for developing and designing pages in the project of eliteLMS.

During the period of his internship program with us, he had been exposed to different processes and was found diligent, hardworking, and inquisitive.

We wish his every success in his career.

For Eejak Technologies Pvt Ltd

Ms Ankita Gaonkar
Authorized Signatory

Eejak Technologies Private Limited
H-134, Sector-63, Noida-201301(U.P.)
Phone # 0120-428-4621 | Website - www.eejak.com

ACKNOWLEDGEMENT

I express my sincere gratitude to everyone who has been instrumental in making my internship at Eejak Technologies Private Limited. a valuable and enriching experience.

I am deeply thankful to Mr. Praveen Kushwaha, the Chief Executive Officer at Eejak, for believing in my capabilities and offering me this incredible opportunity. Your trust and confidence in me have been a source of motivation.

I extend my gratitude to Ms. Ankita Gaonkar, who interviewed me and provided me with the opportunity to join Eejak. Your belief in my potential has been truly inspiring.

I am grateful to Prof. Ramrao Wagh for his mentorship and valuable insights throughout my academic and professional journey.

Furthermore, I would like to thank the entire team at Eejak for their collaboration, support, and for creating a conducive learning environment. Your collective efforts have contributed significantly to my learning and development.

Last but not least, I am thankful to my classmates in college for their encouragement and support.

I am truly fortunate to have had such a supportive network of individuals who have helped shape my internship experience at Eejak. Thank you all for your unwavering support and guidance.

EXECUTIVE SUMMARY

Eejak Technologies is at the forefront of transforming the web development landscape with its advanced use of React and Angular frameworks. This innovative company simplifies the development process, offering significant time and cost savings while delivering seamless user experiences. Eejak Technologies' overarching mission is to streamline web development and enhance digital experiences through its sophisticated use of cutting-edge web technologies.

During my tenure as a Senior Software Engineer, I assumed a pivotal role in various critical projects essential to Eejak Technologies' growth and success. These encompassed a spectrum of tasks, including but not limited to, spearheading comprehensive improvements to the main website, developing an intuitive internal support portal, swiftly addressing production bugs to ensure system stability, integrating dynamic APIs to enhance communication efficiency, and architecting a robust learning management system for seamless customer interaction. Each task presented unique challenges, from balancing user-centric design considerations to tackling intricate technical issues with agility and precision. Through these endeavors, I not only expanded my technical prowess but also cultivated a deeper understanding of the intricate interplay between software development and business objectives.

My internship at Eejak Technologies served as an invaluable learning journey, providing me with exposure to a myriad of advanced concepts and technologies. I delved into the intricacies of React and Angular's advanced features, gaining proficiency in building dynamic user interfaces with enhanced functionality. Embracing the microservices architecture, I acquired insights into scalable and resilient software design principles, essential for managing complex systems effectively. Navigating the vast ecosystem of Amazon AWS, I honed my skills in deploying and managing cloud-based infrastructure, ensuring optimal performance and reliability.

Furthermore, my internship afforded me the opportunity to master the nuances of handling date and time in software applications, a critical aspect often overlooked but indispensable for maintaining data integrity and consistency. Embracing the principles of effective version control with Git, I streamlined collaborative development processes, enabling seamless integration and deployment of code changes. Leveraging server-side events, I gained proficiency in implementing real-time communication channels, essential for building responsive and interactive applications

Additionally, my internship journey encompassed the integration of micro-frontend architecture, empowering me to harness the power of modular and scalable web development for more flexible and resilient applications. Embracing rigorous A/B testing methodologies, I gained insights into data-driven decision-making, enabling me to iterate and optimize software solutions based on empirical evidence and user feedback.

Despite the multifaceted challenges encountered, including the inherent complexity of managing multiple concurrent tasks and the demand for rapid learning and research, my internship at Eejak Technologies was a profoundly enriching experience. It equipped me with the adaptability and resilience required to thrive in dynamic work environments. The mentorship provided was instrumental in guiding my development and enhancing my skills. Moving forward, I am confident that the insights gained and skills honed during my tenure at Eejak Technologies will continue to serve as a strong foundation for my professional growth and contributions in the field of software engineering.

CHAPTER 01 : ORGANIZATION/COMPANY

1.1 BIRDS-EYE-VIEW

Eejak Technologies emerged as a pioneering force in the realm of web development, spearheading a transformative shift towards advanced, client-focused digital solutions. As a leading provider of innovative web development services, Eejak stands at the forefront of technological advancement, redefining the digital landscape for businesses across various sectors.

At the core of Eejak's ethos lies a steadfast commitment to simplifying web development, delivering scalable and dynamic solutions that cater to a diverse range of clients. With a significant presence in the United States, Eejak has garnered a reputation for excellence, particularly in the development of Learning Management Systems (LMS). These systems are extensively used across educational institutions and corporate training programs, enhancing the learning experience with intuitive and user-friendly interfaces.

Expanding its horizons, Eejak is now extending its cutting-edge services to India, with strategic operations centered in Noida and Goa. This expansion aims to tap into the burgeoning tech landscape of India, offering state-of-the-art web development solutions tailored to meet the unique needs of the Indian market.

Eejak's mission is to deliver exceptional web development services that simplify digital experiences and drive business growth. By leveraging advanced frameworks like React and Angular, Eejak provides clients with seamless, efficient, and cost-effective solutions. These technologies not only streamline development processes but also translate into substantial savings and enhanced performance for clients.

Facilitating seamless digital transformations, Eejak caters to a wide spectrum of applications, spanning enterprise solutions, educational platforms, e-commerce, and more. By harnessing the power of modern web technologies, Eejak empowers businesses with dynamic and interactive digital interfaces that foster unparalleled user experiences and operational efficiency.

Above all, Eejak is driven by a mission to shape the future of digital landscapes, enriching lives through efficient, sustainable, and user-friendly digital environments. By prioritizing the success and well-being of businesses and their customers, Eejak endeavors to create intelligent digital spaces that enhance quality of life and contribute to a more connected world.

1.2 PRODUCTS/SERVICES

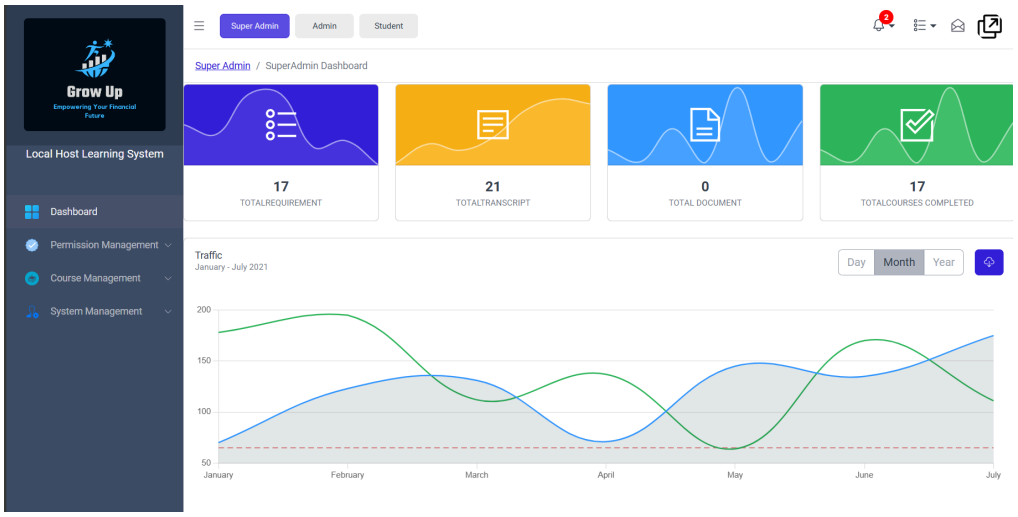
Eejak Technologies excels in delivering web development projects that leverage advanced technology and innovative solutions to meet the evolving needs of their clients. Their flagship Learning Management System (LMS), built with the React framework, is designed to enhance user engagement and learning efficiency. Featuring an intuitive interface, the LMS enables seamless course creation, management, and delivery. Educators benefit from tools to develop interactive modules integrating multimedia content, quizzes, and collaborative features. Robust analytics provide insights into learner progress and engagement, empowering institutions to refine their educational strategies. Real-time notifications, discussion forums, and interactive features foster a community-driven learning environment, enhancing collaboration among students and instructors.

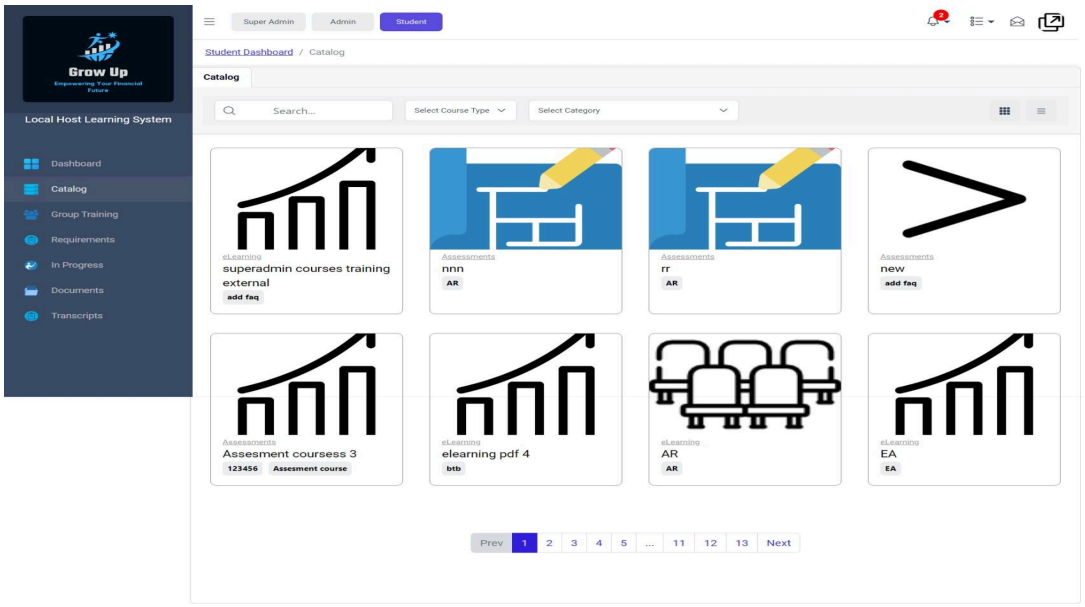
Eejak Technologies' Criminal Records Management System (CRMS), developed with Angular, meets the needs of law enforcement and judicial agencies. Offering advanced search, filtering, and detailed analytics, it facilitates informed decision-making and seamless interdepartmental collaboration. Integrated with other databases, the CRMS ensures comprehensive data management with stringent security measures. Together with their Learning Management System (LMS), Eejak demonstrates a commitment to scalable, secure, and efficient web solutions, driving operational excellence and innovation in digital environments.

1.2.1 Elite LMS (React)

The React-based LMS is highly customizable, allowing institutions to tailor the platform to their specific requirements. This adaptability ensures that the system can support a wide range of educational models, from traditional classroom settings to remote and hybrid learning environments. The platform's modular architecture enables the integration of various third-party tools and plugins, enhancing its functionality and providing a comprehensive learning ecosystem. Features such as real-time notifications, discussion forums, and collaborative tools foster an interactive and community-driven learning environment, making education more accessible and engaging for students.

Security and scalability are at the forefront of Eejak's LMS. The platform employs state-of-the-art security measures to protect sensitive data, ensuring compliance with global standards. Its scalable design allows educational institutions to accommodate an increasing number of users without compromising performance. By leveraging the capabilities of React, the LMS delivers a high-performance, reliable, and user-friendly solution that supports the academic goals of institutions across the United States and beyond. Eejak's commitment to continuous improvement and innovation ensures that their LMS remains at the cutting edge of educational technology.





1.2.2 Criminal Records Management System (Angular)

Eejak Technologies' Criminal Records Management System (CRMS) is an advanced solution developed using Angular, aimed at streamlining and enhancing the management of criminal records for law enforcement and judicial bodies. With its sophisticated search and filtering capabilities, the CRMS allows quick access to critical information, empowering law enforcement personnel to perform their duties with greater accuracy and efficiency, ultimately contributing to improved public safety and justice administration.

The system's robust features include detailed analytics for insights into crime patterns, seamless integration with other databases, and rigorous security protocols to safeguard sensitive information. Utilizing advanced encryption and access controls, the CRMS ensures data integrity and confidentiality, while its scalable architecture accommodates growing data volumes and user numbers, making it a future-proof solution for law enforcement agencies

1.3 SECTIONS WITHIN THE ORGANIZATION

Within any organization, the distribution of responsibilities and functions across various departments is crucial for its smooth operation. Each department plays a specific role in contributing towards the overall success of the company. In this subchapter, we will explore the different sections within our organization, categorizing them into technical (tech) and non-technical (non-tech) fields. This classification will help in understanding the diverse nature of tasks performed across various departments. I will provide insights into both tech and non-tech departments, shedding light on their functions and collaborations within the organization.

1.3.1 Technical Departments

a. Firmware Team

The Firmware Team is responsible for designing and developing embedded software that controls the hardware components of our products. They work closely with the Hardware R&D department to ensure compatibility and efficiency. Their tasks include coding, testing, debugging, and optimizing firmware for devices.

b. Software Development

The Software Development department, where I work, is responsible for creating and maintaining software applications tailored to meet the needs of our customers. This includes coding, testing, debugging, and implementing new features or enhancements. Collaboration with other departments such as Firmware, Integration, and QA is vital to ensure software compatibility and quality.

c. QA (Quality Assurance)

QA teams are tasked with ensuring the quality of both hardware and software products. They develop and execute test plans, identify bugs or issues, and work closely with development teams to resolve them. Their aim is to deliver high-quality products that meet customer expectations.

1.3.2 Non-Technical Departments

a. Accounts

The Accounts department manages financial transactions, budgeting, invoicing, and payroll. They ensure compliance with accounting standards and regulations, providing financial insights and reports to aid decision-making.

b. HR & Admin

The HR & Admin department oversees recruitment, employee relations, training, and administrative tasks. They ensure compliance with labor laws, maintain employee records, and foster a positive work environment.

c. H/W Delivery

Hardware Delivery teams manage the logistics and distribution of physical products. They coordinate with suppliers, warehouses, and shipping carriers to ensure timely delivery of hardware components to customers or other departments.

d. Marketing

Marketing departments are responsible for promoting products and services to attract customers and drive sales. They develop marketing strategies, conduct market research, and execute campaigns across various channels to increase brand visibility and awareness.

CHAPTER 02 : TASK(S) HANDLED

During my internship at Eejak Technologies, I assumed diverse responsibilities as a software developer, primarily centered around web development utilizing React and Node.js technologies.

A significant portion of my internship involved the development of a comprehensive call management system, where I played a crucial role in both frontend and backend functionalities. Leveraging React.js for frontend development, I ensured the creation of intuitive and responsive user interfaces. Simultaneously, backend development using Node.js facilitated efficient server-side logic implementation. Integration of real-time communication features significantly enhanced user experience and streamlined call handling processes.

In addition, I played a pivotal role in promptly addressing production issues by implementing timely bug fixes. Through rigorous debugging and testing, I identified root causes of issues and collaborated with cross-functional teams to prioritize and resolve critical bugs, thereby ensuring system reliability and performance.

I'll briefly explain some of the tasks that I worked on during my internship period, focusing more on the Learning management feature, because it was the most interesting thing that I worked on during the internship and also because I was responsible for both frontend and backend development of this service.

2.1 CONTRIBUTION TO API OPTIMIZATION

During my internship, I had the opportunity to contribute significantly to optimizing the company's API performance, particularly focusing on improving response handling across different interfaces.

One pivotal aspect of this optimization effort involved creating separate files for response handling in the superadmin, admin, and student interfaces. By segregating response handling logic, we aimed to streamline code management and enhance scalability. This allowed for more efficient management of API responses tailored to the specific needs and functionalities of each interface.

Additionally, I collaborated closely with cross-functional teams to identify areas for improvement and implement optimizations. This included conducting thorough performance audits to identify bottlenecks and inefficiencies in API response handling. Through this process, we implemented various optimization techniques such as code minification, response caching, and asynchronous loading, resulting in significant improvements in response times and overall performance.

Throughout the optimization process, effective communication and collaboration were

crucial. I worked closely with developers, product managers, and UI/UX designers to gather requirements, brainstorm ideas, and ensure alignment with the company's objectives. Regular updates and feedback sessions helped to ensure that the optimization efforts were on track and delivered tangible value to the company and its users.

2.2 MICRO-FRONTED ARCHITECTURE IMPLEMENTATION

During my internship, I had the privilege to work on implementing a Micro-Frontend Architecture for segregating the company's applications—Superadmin, Admin, and Student—into independent web applications. Leveraging React.js for the frontend development, we aimed to achieve modularity and scalability in our application architecture.

2.2.1 Purpose and Functionality

The primary objective of implementing the Micro-Frontend Architecture was to modularize the company's applications, allowing for independent development and deployment of frontend features by different teams. By breaking down the monolithic frontend into smaller, independently deployable components, we aimed to improve development speed, maintainability, and scalability across the Superadmin, Admin, and Student applications.

2.2.2 Features

a. Component-Based Architecture

Utilizing the Micro-Frontend Architecture, each application was divided into smaller, self-contained modules or micro-frontends. Each micro-frontend was responsible for a specific feature or functionality within its respective application, allowing teams to develop, test, and deploy their components independently.

b. Seamless Integration

Through the use of techniques such as lazy loading and asynchronous module loading, the micro-frontends seamlessly integrated with each other within their respective applications. This allowed for faster page loading times and reduced overhead, enhancing the overall performance of the Superadmin, Admin, and Student applications.

c. Team Collaboration

The implementation of the Micro-Frontend Architecture promoted greater collaboration among team members, as each member was responsible for developing and maintaining specific micro-frontends within their application. This distributed ownership encouraged a sense of accountability and pride in the work produced, resulting in higher quality and more innovative solutions.

2.2.3 Continuous Improvement

With the Micro-Frontend Architecture in place, the Superadmin, Admin, and Student applications were able to evolve rapidly to meet the changing needs of the organization. New features could be developed and deployed independently within each application, allowing for faster iteration cycles and continuous improvement of functionality and user experience.

Advantages of Micro-Frontend Architecture

1. Modularity

Micro-Frontend Architecture allows breaking down large, monolithic frontend applications into smaller, more manageable modules. Each module or micro-frontend can be developed, tested, and deployed independently, promoting modularity and enabling teams to work on different parts of the application simultaneously.

2. Scalability
By dividing the frontend into smaller, self-contained modules, Micro-Frontend Architecture facilitates scalability. Teams can scale individual modules based on demand, without impacting the entire application. This allows for more efficient resource utilization and ensures that the application can handle increased traffic and user load effectively.

3. Independent Development and Deployment
With Micro-Frontend Architecture, teams have the autonomy to develop and deploy their micro-frontends independently. This enables faster iteration cycles and promotes agility, as teams can release new features or updates without waiting for other teams' schedules or dependencies.

4. Technology Agnosticism
Micro-Frontend Architecture allows teams to use different technologies, frameworks, or programming languages for developing individual micro-frontends. This technology agnosticism promotes flexibility and innovation, as teams can choose the most suitable tools for their specific requirements, without being constrained by the technology choices of other teams.

5. Improved Performance
Micro-Frontend Architecture can enhance the performance of frontend applications by enabling lazy loading and asynchronous loading of modules. Only the necessary modules are loaded when needed, reducing initial load times and improving overall page responsiveness. Additionally, smaller codebases and reduced dependencies can lead to faster rendering and improved user experience.

6. Team Autonomy and Ownership
Micro-Frontend Architecture encourages team autonomy and ownership, as each team is responsible for developing and maintaining specific micro-frontends. This distributed ownership fosters a sense of accountability and pride in the work produced, resulting in higher quality and more innovative solutions. Additionally, teams can iterate and innovate more freely, leading to continuous improvement and innovation in the application.

2.2.4 Limitations of Mono-lithic Architecture Compared to Micro-Frontend Architecture:

1. Monolithic Nature
Traditional monolithic architectures often result in large, tightly coupled codebases where all components are tightly integrated. This can lead to challenges in scalability and maintainability, as any changes or updates to one part of the application can potentially impact the entire system.

2. Limited Agility
In monolithic architectures, development, testing, and deployment are typically done as a single unit. This lack of modularity makes it challenging to implement agile development practices such as continuous integration and continuous deployment (CI/CD), resulting in slower iteration cycles and reduced agility in responding to changing business requirements.

3. Technology Lock-in
Monolithic architectures often require teams to use the same technology stack for the entire application. This can lead to technology lock-in, where teams are unable to adopt new technologies or frameworks that may better suit their specific requirements. It also limits innovation and flexibility, as teams are constrained by the technology choices made early in the development process.

4. Limited Scalability
Scaling monolithic applications can be challenging, as the entire application must be scaled horizontally to accommodate increased traffic or user load. This can lead to inefficiencies in resource utilization and increased infrastructure costs. Additionally, scaling individual components or features independently is not feasible, making it difficult to optimize performance and cost-effectiveness.

2.2.5 Key Considerations for Effective Micro-fronted Development

1. Clear Boundaries and Contracts
Establish clear boundaries between micro-frontends to ensure separation of concerns and minimize dependencies. Define well-defined contracts or APIs for communication between micro-frontends to facilitate seamless integration and interoperability.

2. Consistent Design and UX Guidelines
Maintain consistency in design and user experience across all micro-frontends to provide a cohesive user experience. Establish and adhere to design and UX guidelines to ensure that users have a seamless experience as they navigate between different parts of the application.

3. Independent Development and Deployment
Enable teams to develop and deploy their micro-frontends independently to promote agility and autonomy. Implement automated testing and deployment pipelines for each micro-frontend to streamline the development process and minimize downtime.

4. Monitoring and Performance Optimization
Implement monitoring and performance optimization strategies to ensure the reliability and scalability of micro-frontends. Monitor key metrics such as response times, error rates, and resource utilization to identify and address performance bottlenecks proactively. Employ techniques such as lazy loading and code splitting to optimize loading times and enhance user experienCHAPTER 03 : LEARNING

During my internship, I gained invaluable experience and deepened my knowledge in several key areas that have significantly contributed to my development as a software developer. These learnings spanned advanced technical concepts, practical tool usage, and important non-technical skills crucial for professional growth in a tech company.

On the technical front, I delved into advanced concepts of React, which enhanced my ability to build dynamic and responsive user interfaces. I explored the architecture and implementation of microservices, which provided me with insights into designing scalable and maintainable systems..

Effective version control using Git became second nature, improving my collaboration and code management capabilities.

Beyond technical skills, I acquired non-technical insights essential for thriving in a tech environment. These included strategies for effective communication, teamwork,

and professional development, which are all critical for long-term success in the industry.

CHAPTER 03 : LEARNINGS

3.1 REACT ADVANCED CONCEPTS

At Eejak, we majorly use React for frontend development so I had the opportunity to dive deep into advanced concepts of React, which significantly enhanced my proficiency in this powerful JavaScript library. Building upon my foundational knowledge, I explored several key areas that are critical for developing complex, high-performance web applications. I have listed down some of this key concepts that I had to learn in details during my internship period.

3.1.1 Component Lifecycle

Understanding the component lifecycle in React was crucial for optimizing the performance and behavior of our application. I delved into lifecycle methods such as componentDidMount, componentDidUpdate, and componentWillUnmount, learning how to manage state and side effects effectively. This knowledge allowed me to implement efficient data fetching, resource cleanup, and performance tuning within my components.

3.1.2 State Management

Managing state in large applications can be challenging, and I explored various strategies to handle this complexity. I gained hands-on experience with the Context API, which helped in passing data through the component tree without having to prop-drill at every level. Additionally, I worked with popular state management libraries like Redux and Zustand, understanding how to maintain a global state and ensure consistency across the application.

3.1.2 Hooks

I learned to use built-in hooks like useState, useEffect, useContext, and useReducer to manage state and side effects in a more declarative manner. Custom hooks also became an essential part of my toolkit, enabling me to encapsulate reusable logic and keep my components clean and focused.

3.1.2 Performance Optimization

Performance is a key consideration in web development, and I delved into techniques for optimizing React applications. I learned about code splitting using React.lazy and Suspense to load components on demand, reducing the initial load time. Memoization techniques with React.memo and useMemo helped in preventing unnecessary re-renders and improving the efficiency of the application. Additionally, I explored the use of tools like the React Developer Tools and profiling features to identify and address performance bottleneck

3.2 USING GIT EFFECTIVELY

During my internship, mastering Git for version control was crucial to maintaining a smooth workflow and collaboration within our development team. Git allowed us to track changes, manage code versions, and collaborate efficiently.

In our company, we followed a specific branch naming convention to ensure consistency and clarity: DEVELOPER_FEATURE_DATE. This convention helped in easily identifying who worked on which feature and when it was created. Every new feature branch was checked out from the main branch. For example, if a developer named Aiwin was working on a login feature on May 5th, the branch would be named Aiwin_1505.

We managed multiple environments (development, testing, staging, production), each with its respective base branch. Here's how our workflow operated:

1. **Feature Development in Dev Environment:**

   Check Out a New Branch:

   ```
   git checkout main git pull
   origin main
   git checkout -b Aiwin_1505
   ```

   Develop and Commit Changes:
   Regular commits are made to the feature branch, documenting the progress and changes:

   ```
   git add .
   git commit -m "Implemented new feature"
   ```

   Reset to a specific branch:
   Resetting to a previous branch:

   ```
   git reset hard -(commit_id).
   ```

   Push the Branch:
   After development and local testing, the branch is pushed to the remote repository:

   ```
   git push
   ```

   Merge to Development Base Branch:
   Once basic testing is completed by the developers, the feature branch is merged with the dev branch:

```
git checkout dev
git mergeAiwin_1505  git push
```

2. **Testing in Test Environment:**

   Merge to Testing Base Branch:
   After merging with the dev branch, the feature is moved to the testing environment by merging the feature branch with the test branch:

```
git checkout test
git merge  Aiwin_1505 git push
```

   QA Testing:
   QA conducts thorough testing on the test branch. There could be several new features in the testing environment which are not yet deployed to production.

3. **Preparation in Staging Environment:**

   Create Staging Branch:
   A single feature approved by QA is moved to the staging environment. A staging branch is created from the main branch and is merged with the feature branch:

```
git checkout main git pull
origin main
git checkout -b Aiwin_1505 git merge  Aiwin_1505 git
push
```

   QA Testing on Staging:
   QA conducts another round of testing on the staging branch. This branch represents the main branch plus the QA-approved feature.

4. **Release to Production:**

   Merge Staging to Main
   Once the staging branch is approved, it is merged into the main branch by raising a pull request (PR). The PR is then approved and merged with main, so now the feature is released to production.

3.3 LIBRARIES LEARNT

During my time at Eejak, I immersed myself in mastering various React libraries, which significantly enriched my expertise in frontend development. Exploring these libraries allowed me to delve into advanced concepts and techniques essential for crafting dynamic and efficient web applications. Below are some of the key libraries I extensively learned during my internship:

3.3.1 Framer Motion

Understanding Framer Motion was pivotal for creating fluid and engaging animations within our React applications. I delved into its animation APIs and features, mastering techniques for creating smooth transitions and interactive user interfaces. This knowledge empowered me to enhance the visual appeal and user experience of our applications through captivating animations.

3.3.2 CoreUI

Exploring CoreUI provided me with valuable insights into building responsive and feature-rich user interfaces. I learned to leverage its components and layouts effectively to create intuitive and visually appealing dashboards and admin panels. This library enabled me to streamline the development process and deliver polished UI designs that met our project requirements.

3.3.3 PrimeReact

Mastering PrimeReact was instrumental in developing robust and scalable frontend solutions. I gained proficiency in utilizing its extensive collection of UI components and utilities to accelerate development and ensure consistency across our applications. PrimeReact empowered me to create professional-grade interfaces with ease, enhancing both productivity and user satisfaction.

3.3.4 Ant Design

Delving into Ant Design enabled me to leverage its comprehensive set of design components and patterns to create modern and responsive user interfaces. I learned to customize and integrate Ant Design components seamlessly into our React applications, achieving a cohesive and polished design language. This library facilitated rapid prototyping and development, allowing us to deliver high-quality UI solutions efficiently.

CHAPTER 04 : CHALLENGES

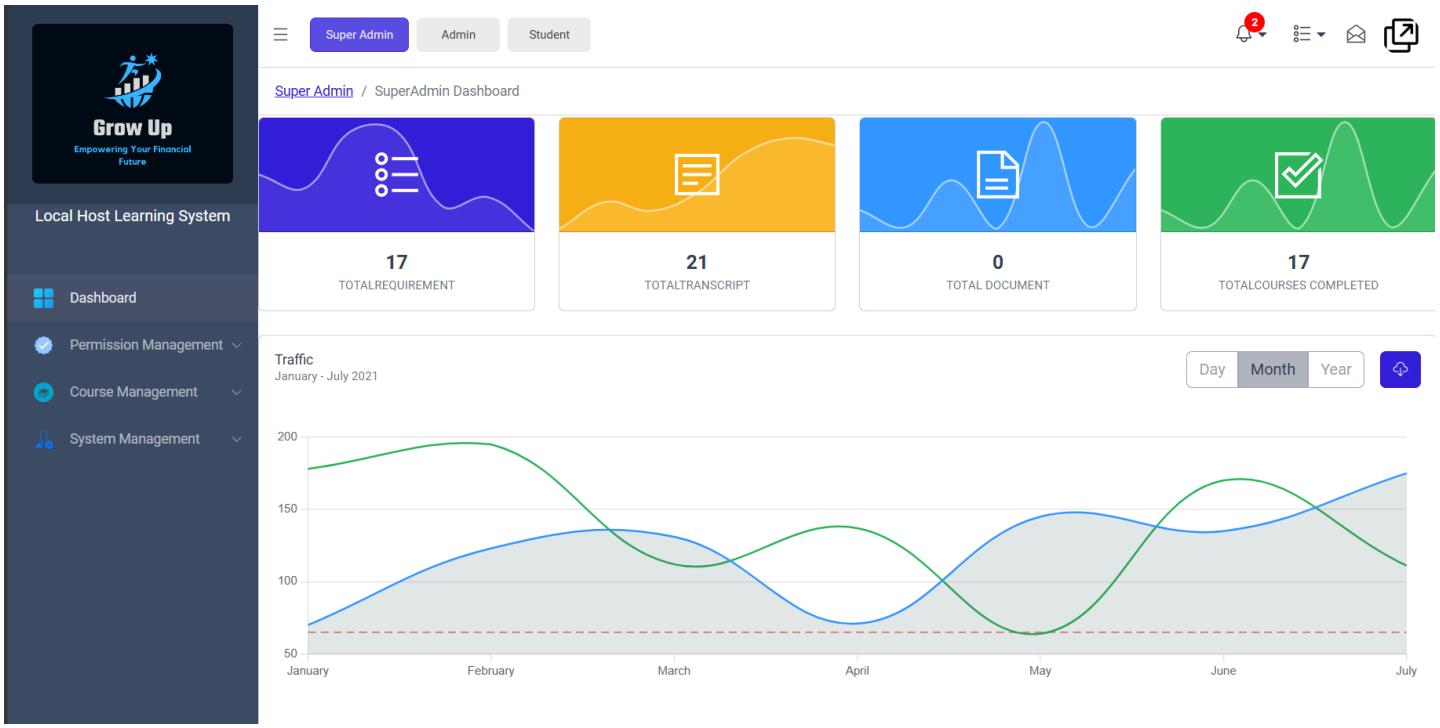
4.1 Understanding the Specifics

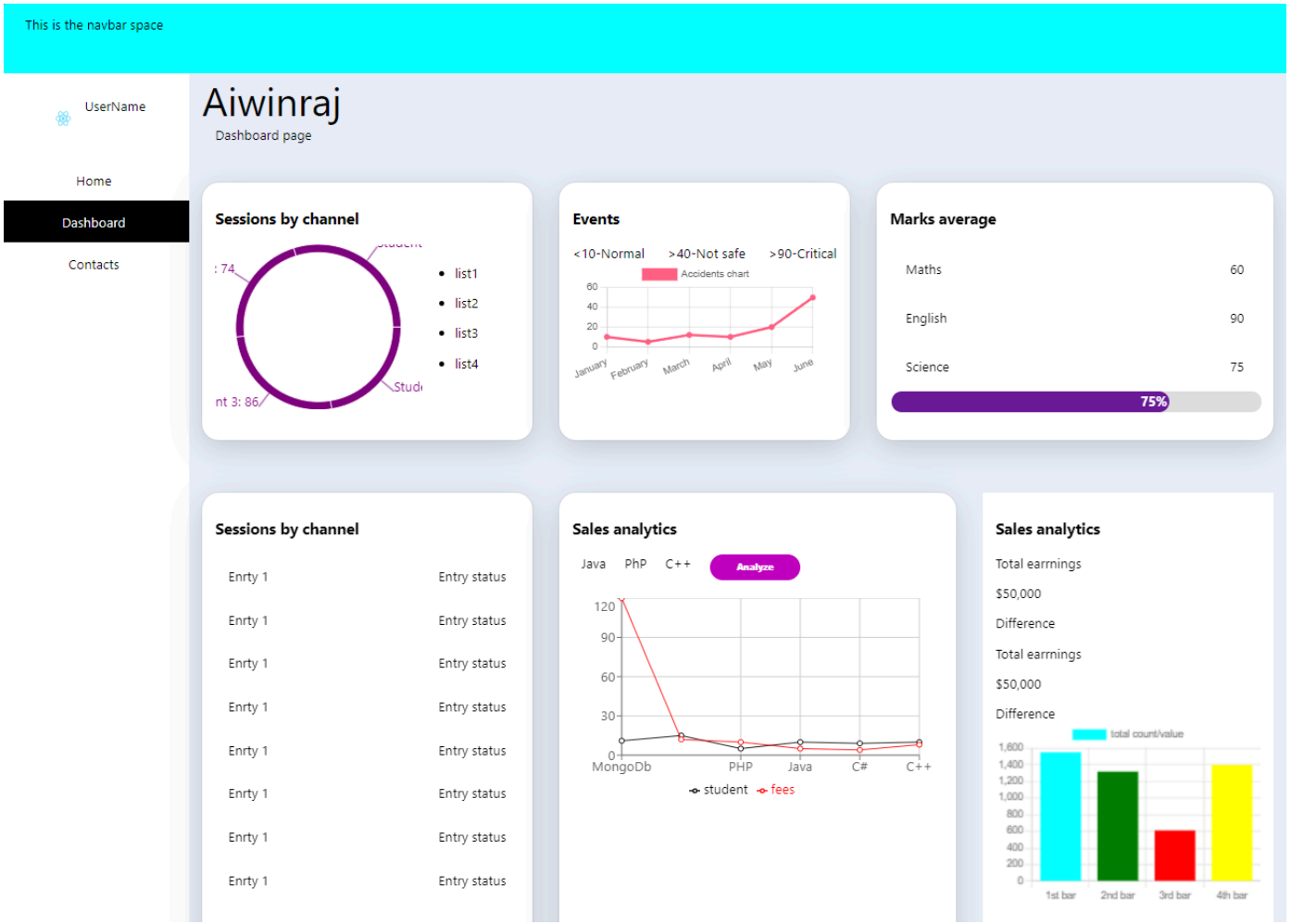
In my company, I faced a significant challenge with my manager's communication. Often, my manager would assign tasks without providing complete clarity. When I sought further explanation, the responses remained vague, leaving me uncertain about the expectations. Despite my best efforts to interpret the instructions and complete the tasks, my manager frequently expressed dissatisfaction with the results, as they did not align with the uncommunicated specifics. This situation hindered my ability to deliver the expected outcomes efficiently. Ultimately, it highlighted the need for more precise communication and better mutual understanding in our team.

4.2 Lack of Employees

In the same company, there was a high turnover rate, with employees joining and leaving within intervals of 2-3 weeks. The reasons for this frequent turnover were unclear, but it significantly affected my collaborative workflow. Each new employee needed time to catch up with the project, which disrupted the continuity and efficiency of our work. The constant onboarding process diverted resources and attention from project tasks. This situation created challenges in maintaining consistent progress and meeting deadlines. As a result, it underscored the importance of team stability for effective project execution.

4.3 Appendix I : Samples of the work done




This is the navbar space
UserName
Home
Dashboard
Contacts
Aiwinraj
Dashboard page
Sessions by channel
: 74
nt 3: 86/
list1
list2
list3
list4
Events
<10-Normal
>40-Not safe
>90-Critical
Accidents chart
60
40
20
0
January
February
March
April
May
June
Marks average
Maths 60
English 90
Science 75
75%
Sessions by channel
Enrty 1 Entry status
Enrty 1 Entry status
Enrty 1 Entry status
Enrty 1 Entry status
Enrty 1 Entry status
Enrty 1 Entry status
Enrty 1 Entry status
Enrty 1 Entry status
Sales analytics
Java
PhP
C++
Analyze
120
90
60
30
0
MongoDb
PHP
Java
C#
C++
student
fees
Sales analytics
Total earrnings
$50,000
Difference
Total earrnings
$50,000
Difference
total count/value
1,600
1,400
1,200
1,000
800
600
400
200
0
1st bar
2nd bar
3rd bar
4th bar