VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science Engineering & Information Systems (SCORE)
Department of Computer Applications

Fall Semester 2023-24

ITA6099 – Off-Campus Internship

Master Thesis Report

Reg. No: 22MCA0312

Student Name: Sayyed Abid Miya Nazim Miya

Guide Name: Prof. Thanga Mariappan L

Company Name: Hive ADS Media

Project:

1. Enterprises Management System for Company (PHP, MySQL)
2. React and Express Web Application Project
3. WordPress Website

Front End: JavaScript, React, Tailwind CSS, HTML, CSS

Back End: PHP, MySQL,Node.js, Hostinger

Tools: Git, VS Code

CMS: WordPress

ACKNOWLEDGEMENT

It is my pleasure to express with a deep sense of gratitude to my master's thesis guide, **Dr. Thanga Mariappan**, School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, for his constant guidance, continual encouragement, and understanding. More than all, he taught me patience in my endeavour. I would also like to extend my heartfelt thanks to my team leader**, Mr. Pritish Wakhare**, for his invaluable support, mentorship, and collaboration throughout the duration of my internship. His expertise, guidance, and dedication have played a pivotal role in shaping my learning experience and contributing to the success of our projects

"I would like to express my heartfelt gratitude to **Dr. G Viswanathan,** Chancellor; **Mr. Sankar Viswanathan,** Vice President; **Dr. Sekar Viswanathan,** Vice President; **Dr. G V Selvam,** Vice President; **Dr. V. S. Kanchana Bhaaskaran,** Vice Chancellor; **Dr. Partha Sharathi Mallick,** Pro-Vice Chancellor; **Dr. Jayabarathi T,** Registrar **and Dr. Sumathy S,** Dean of School of Computer Science Engineering and Information Systems, for providing me with an enriching environment to work in and for their inspirational guidance throughout the tenure of the course. "In a jubilant mood, I express ingeniously my whole-hearted thanks to **Dr E Vijayan** Associate Professor Senior & Head, Department of Computer Applications, MCA Project Coordinator, all teaching staff and members working as limbs of our university for their not-self-centred enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. Last, but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore

Date: **Sayyed Abid Miya Nazim Miya**

Table of Contents

Introduction

Three distinct projects were undertaken, each built on different technology stacks.

The first project involved developing an Enterprise Management System. This system is designed to efficiently track employee activities, tasks, and attendance, providing real-time insights into their workflow and productivity.

The second project cantered around enhancing the company's personnel website, which was initially built on WordPress. This endeavour focused on improving the website's design, incorporating various forms, and enhancing its overall functionality.

In the final project, I worked on a web application built using React and Express, catering to a college client. My responsibilities included addressing UI issues, enhancing existing functionalities, and introducing new features to improve user experience and efficiency.

These projects provided valuable hands-on experience in web development and served as the basis for my master's thesis, which delves into the practical aspects of client-centric web application development and the challenges encountered therein.

Enterprises Management System

1. Design principle

- **Overview:**

  Enterprises Management System (EMS) is designed with the principle of modularity in mind, allowing scalability and maintenance as the organization grows. This approach ensures that the EMS can handle and manage all employees, even in the thousands, by organizing tasks efficiently. The design is not only suited for current functionality but also for future enhancements and updates.

- **Modularity Benefits:**

  This design enables developers to work on individual modules independently, promoting parallel development and reducing interdependence among modules. Developers can work freely on their assigned modules without worrying about affecting others. Git tools are used for source code management and collaboration, facilitating coordination among developers working on the same module.

- **Scalability and Maintainability:**

  The modular design principle is crucial for scalability. As the EMS grows in features and complexity, additional modules can be easily added, and new libraries can be integrated seamlessly to enhance functionality. Scalability ensures the system can accommodate multiple employee logins and track each employee's tasks in real-time. Maintenance becomes more manageable, as specific parts can be updated without affecting the entire system

2. Technology Stack

A We use core PHP for server-side scripting and business logic, MySQL for database management, and PHPMyAdmin for data administration. Tailwind CSS is employed for UI, including HTML and CSS in some parts.

PHP:

PHP (Hypertext Preprocessor) serves as the primary language for both server-side scripting and business logic. Its versatility allows easy embedding with HTML and Tailwind CSS to generate dynamic content. The extensive community support, numerous libraries, and readily available resources on platforms like Stack Overflow contribute to efficient and rapid development.

PhpMyAdmin for Database Administration:

While MySQL shell can control and administer data, PhpMyAdmin streamlines the process, saving time and allowing a focus on business logic. It simplifies tasks like creating databases, making modifications, and maintenance, making it particularly useful in a local environment like XAMPP for testing the EMS web app.

Tailwind CSS:

Tailwind CSS is chosen for its rapid development features. Transitioning from a HTML and CSS-only EMS, Tailwind CSS facilitates fast development without affecting the backend, making it a preferred choice within the community.

3. Testing and Deployment

Unit Testing
Unit Testing involves testing individual units or components of a software application in isolation to ensure they function as expected. It focuses on verifying the correctness of small, atomic sections of code, such as functions or classes. Unit tests are automated and help detect defects early in the development process, ensuring code quality and facilitating refactoring.

Integration Testing
Integration testing verifies the interactions and interfaces between different units or modules of a software application when they are combined and tested as a group. It ensures that the integrated system functions correctly by testing data flow, communication between components, and overall system behaviour. Integration testing helps identify issues that may arise when integrating different parts of the application, ensuring seamless operation of the entire system.

PHPUnit Testing library

For testing React components and functionality, we are using the PHPUnit Testing Library and writing a combination of unit and integration tests for the web app. Unit testing ensures individual components work correctly, while integration testing checks the interaction of multiple components.

Deployment

- **Hosting platform**

For the hosting platform, we are using Hostinger.

4. Challenges Faced and solution Implemented

Encountering a problem while integrating the Tailwind library provided an opportunity to showcase my problem-solving skills. I also demonstrated my ability to track down a challenging bug - our system, responsible for live-tracking employee tasks, kept logging out repeatedly. We implemented a solution using a WebSocket instead of relying solely on HTTPS for live time tracking. Additionally, I improved the performance of the company website by redesigning it.

Critical Elements:

- Analytical Approach: The use of profiling tools showcased an analytical approach to problem-solving, identifying specific areas impacting system performance. Also, the UI's lack of interactivity was addressed.
- Collaborative Problem-Solving: Addressing a performance bottleneck often requires collaboration. I worked closely with team members to implement optimization techniques, drawing on collective expertise to achieve the best results for the company website and resolving the bug through the use of WebSocket.
- User-Centric Focus: The ultimate goal of problem-solving was to enhance the user experience. This approach ensured that technical solutions aligned with the overarching objective of delivering a high-performing and user-friendly enterprise management system. I suggested using Tailwind CSS for this purpose.

5. Lesson Learned

self-evaluation is one of the most important things for professional growth, and the ability to learn from mistakes is a valuable skill in any work environment. By acknowledging and rectifying errors, not only does it improve my individual competence, but it also contributes to the overall success of projects.

- **Personal Growth:**

  When I was working on an Enterpise Management System, the whole system was written in core PHP and HTML, CSS. So, first, I had to revise the PHP I learned in my academic course.

  Frontend: Then I faced a significant challenge; I had to change the entire UI using Tailwind CSS. First, I had to learn the framework. The standard way of installing Tailwind CSS is using Node, but we were using core PHP, and there is no npm (Node Package Manager) for PHP. So, I had to find another way of integrating it by using Tailwind CLI. Then, I had to design the whole directory again for the project because our idea was to create a UI component so that we could easily integrate and use it in the project.

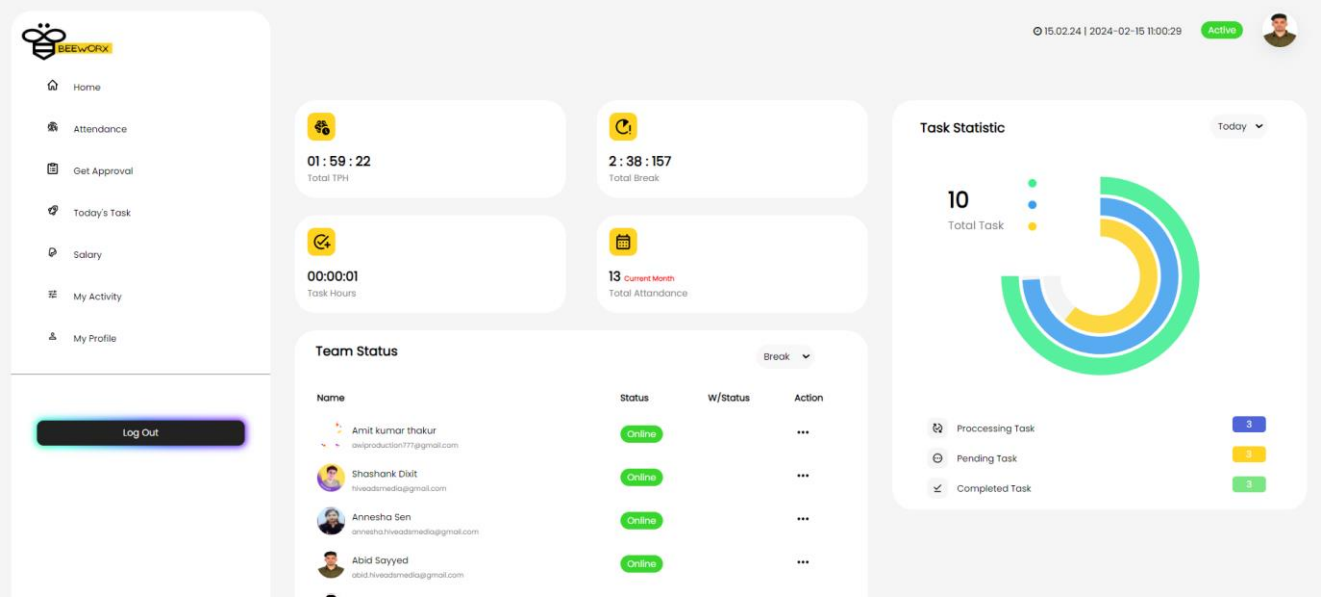
  Backend: This Enterprise Management System was very challenging because it involved live tracking of an employee's activity in real-time. I had to understand various technologies before contributing to the backend. I learned about websockets, the PHP mail library, jQuery for user interaction, and phpMyAdmin for storing data. Before contributing, I fixed various bugs, which helped me understand the entire architecture. I also learned the importance of troubleshooting. Then, I contributed a new feature for the brake system.

- **Adaptability:**

  Recognizing my initial lack of expertise, I proactively engaged in self-learning. Through extensive research and experimentation, this experience underscored the significance of adaptability and the willingness to acquire new skills when faced with novel challenges.

6. Implementation Detail

EMS SYSTEM



DASHBOARD

7. CONCLUSION

The development of the Enterprise Management System serves as a prime example of how the quantity and quality of project work, combined with strict adherence to engineering standards, contribute to the success of a project. The outcome is a reliable, scalable, and secure solution that meets the organization's requirements while aligning with industry best practices. I continue to learn and improve my skills, including programming, communication, collaboration, and expressing ideas, through this experience.

React and Express Web Application Project

1. Design principle

- **Overview:**

  This Node.js web-based project is using React and Express Library is designed with the principle of modularity in mind, allowing scalability and maintenance as the organization grows. This approach ensures that the WebAPP can handle and manage all Student, even in the thousands, by organizing tasks efficiently. The design is not only suited for current functionality but also for future enhancements and updates.

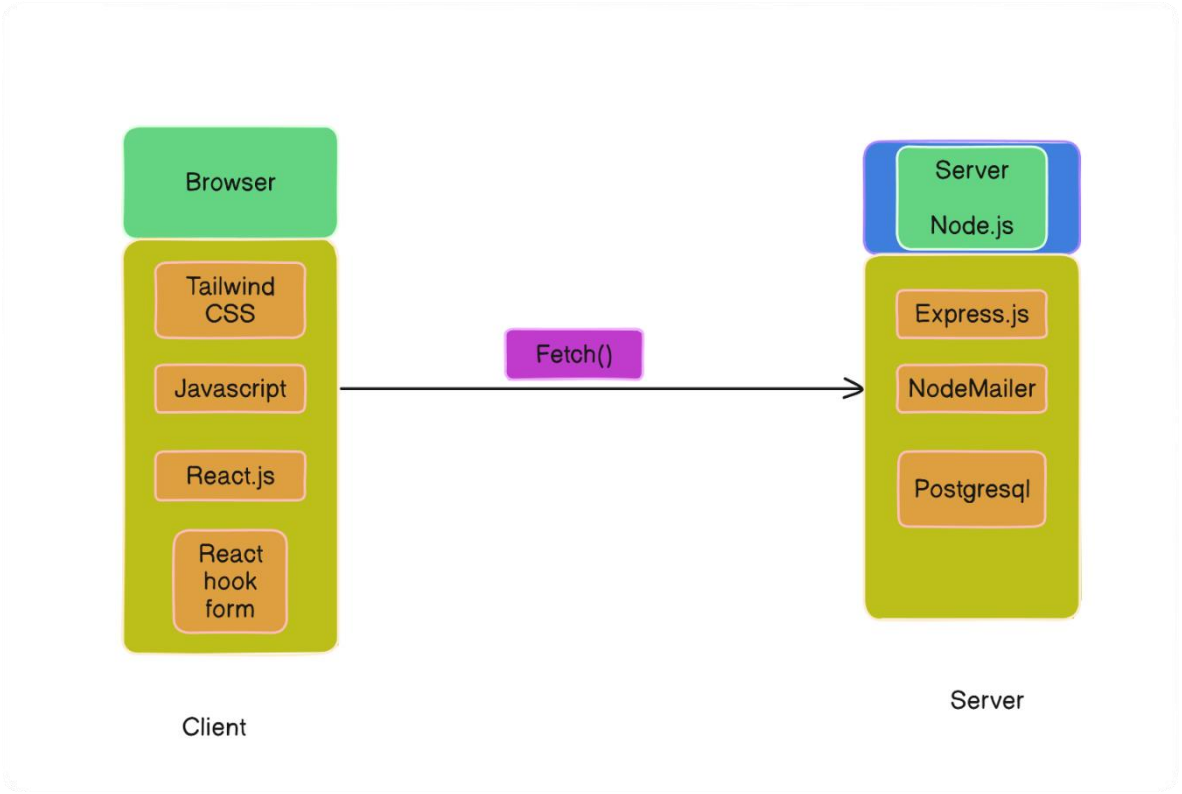
- **Modularity Benefits:**

  This design enables developers to work on individual modules independently, promoting parallel development and reducing interdependence among modules. Developers can work freely on their assigned modules without worrying about affecting others. Git tools are used for source code management and collaboration, facilitating coordination among developers working on the same module.

- **Scalability and Maintainability:**

  The modular design principle is crucial for scalability. As the EMS grows in features and complexity, additional modules can be easily added, and new libraries can be integrated seamlessly to enhance functionality. Scalability ensures the system can accommodate multiple employee logins and track each employee's tasks in real-time. Maintenance becomes more manageable, as specific parts can be updated without affecting the entire system

2. Architecture



3. Technology Stack

- **Overview**

  After the Enterprise System, I have been assigned to another client project. In this project, the client is a college, and we are tasked with fixing some bugs and implementing new features in their existing web application. The web application is built with an Express.js backend and a React frontend, with Tailwind CSS for the user interface.

  JavaScript:

  JavaScript serves as the primary language for both server-side scripting and business logic. Using the same language on both frontend and backend with the help of Node.js greatly facilitates development. The extensive community support, numerous libraries, and readily available resources on platforms like Stack Overflow contribute to efficient and rapid development.

  Frontend

  React.js:

  Serves as the main frontend library for building dynamic web applications systematically. It helps in controlling each part of the webpage dynamically without refreshing the entire page. For this, it uses virtual DOM, making operations efficient.

  Tailwind CSS:

  Tailwind CSS is chosen for its rapid development features and ease of integration with React. It facilitates fast development without affecting the backend, making it a preferred choice within the community.

  Backend

  Express.js:

  we are using Express.js, which connects with the database and sends data to frontend React. It acts as an API for the frontend part. This library helps in connecting with the database easily, building RESTful APIs, and ensuring secure data transfer. Many libraries, such as Nodemailer, are utilized with Express.js.

  PostgreSQL
  For database storage, we are using PostgreSQL, a relational database, with Sequelize as an ORM.

4. Testing and Deployment

Unit Testing
Unit Testing involves testing individual units or components of a software application in isolation to ensure they function as expected. It focuses on verifying the correctness of small, atomic sections of code, such as functions or classes. Unit tests are automated and help detect defects early in the development process, ensuring code quality and facilitating refactoring.

Integration Testing
Integration testing verifies the interactions and interfaces between different units or modules of a software application when they are combined and tested as a group. It ensures that the integrated system functions correctly by testing data flow, communication between components, and overall system behaviour. Integration testing helps identify issues that may arise when integrating different parts of the application, ensuring seamless operation of the entire system.

PHPUnit Testing library

For testing React components and functionality, we are using the PHPUnit Testing Library and writing a combination of unit and integration tests for the web app. Unit testing ensures individual components work correctly, while integration testing checks the interaction of multiple components.

React Testing library

For testing React components and functionality, we are using the React Testing Library and writing a combination of unit and integration tests for the web app. Unit testing ensures individual components work correctly, while integration testing checks the interaction of multiple components.0

Deployment

- **Hosting platform**

For the hosting platform, we are using Hostinger.

5. Challenges Faced and solution Implemented

.

- **Bug Fixing ability:**

  I was tasked with fixing bugs such as page layout issues on mobile, images not displaying correctly, and incorrect data fetching. This improved my ability to identify and solve bugs effectively.

- **Adding New Features:**

  I learned to seamlessly integrate new features into our application, collaborating with teammates and ensuring compatibility with existing code. For instance, I added a feature to display urgent notices on the homepage.

- **Learning the Usage of Correct Tools Based on Situation:**

  I learned to identify and employ the correct tools based on the situation, ensuring optimal productivity and effective problem-solving.

- **Problem-Solving Strategies:**

  I approached challenges with analytical thinking and creativity, systematically analysing root causes and exploring solutions to drive continuous improvement.

  I successfully completed all tasks, receiving credit from my team leader for delivering them early. This experience has enhanced my debugging and development speed and deepened my understanding of React and Node.js lifecycles.

6. Lesson Learned

So, after the Enterprise System, I am assigned a client web application. A client is collecting for which we are handling their complex web app. The web application is built on Express.js backend and react as frontend and Tailwind CSS as UI.

- **Performance:**

So, I was tasked to improve certain UI and fix certain UI bugs, then I was assigned for the React part, where I had to implement some features. I successfully completed all my tasks and was able to deliver them by thoroughly testing and on time, for which I got credit from my team leader for completing so early. So as of now, I have got the hang of all the projects which are under mine, which continually improves my debugging as well as development speed, and I got familiar with the overall of React and Node.js lifecycle.

- **Skill Development:**

Frontend: For frontend, we were using React, so I had to revise the JavaScript first, which I learned in MCA batch, then I also had to go through React concepts. But I have learned some new things like how to use fetch API for fetching the data from the backend, how frontend and backend connect using an API, and also learned instead of writing from scratch if there is some library I can use that also but the condition is I have to thoroughly check if it's correct and open source. I had to build some forms also for which I used the React-Hook form. So overall I learned all the things required to build a frontend in React.

Backend: For the backend part, we were using Express.js. This part was mainly handled by another backend developer, but I also got the chance and permission to see how the backend framework works by working with it with a simple case. Mainly I had to understand all the model architecture and all models which are written with Sequelize which is an ORM for MySQL. Because before fetching the data in the frontend I needed to understand how the model is written. I also got familiar with Nodemailer which sends mail to the users. I have also written some test cases using React Testing Library. So overall after working on a real project, I got how development lifecycle works and how to fix the bug based on criteria and how to collaborate with each other in real-time.

7. CONCLUSION

The development of this web application using reat.js and exprss.js and nojde.js serves as a prime example of how the quantity and quality of project work, combined with strict adherence to engineering standards, contribute to the success of a project. The outcome is a reliable, scalable, and secure solution that meets the organization's requirements while aligning with industry best practices. I continue to learn and improve my skills, including programming, communication, collaboration, and expressing ideas, through this experience.

WordPress based Company personal Website

1. Design principle

A WordPress, as a content management system (CMS), follows several design principles to ensure flexibility, usability, and extensibility. These principles guide the development of WordPress core, themes, and plugins, ensuring consistency and ease of use for users and developers alike. Here are some key design principles of WordPress

1. **Simplicity**: WordPress aims to be easy to use for both beginners and experienced users. The user interface is designed to be intuitive and straightforward, with minimal clutter and complexity. This simplicity extends to the backend interface for content management as well.

2. **Flexibility**: WordPress is designed to be highly flexible and customizable. Users can easily change the appearance of their website by switching themes or modifying theme templates. Additionally, plugins extend the functionality of WordPress, allowing users to add features and customize their websites according to their needs.

3. **Accessibility**: Accessibility is a core principle of WordPress design. The platform strives to ensure that websites built with WordPress are accessible to all users, including those with disabilities. This includes adherence to web accessibility standards, such as WCAG (Web Content Accessibility Guidelines), and providing tools and features to help developers create accessible themes and plugins.

4. **Modularity**: WordPress follows a modular design approach, where different components (such as themes, plugins, and core functionality) are loosely coupled and can be easily extended or replaced. This modularity enables developers to build on top of WordPress without having to modify core files, making it easier to maintain and update websites.

5. **Compatibility**: WordPress is designed to be compatible with a wide range of web hosting environments, browsers, and devices. Themes and plugins are expected to follow best practices to ensure compatibility with different WordPress versions and configurations. This compatibility ensures that websites built with WordPress can reach a broad audience without compatibility issues.

2. Technology Stack

Html / CSS

Serves as the main frontend library for building dynamic web applications systematically. It helps in controlling each part of the webpage dynamically without refreshing the entire page. For this, it uses virtual DOM, making operations efficient.

WordPress CMS

we are using Express.js, which connects with the database and sends data to frontend React. It acts as an API for the frontend part. This library helps in connecting with the database easily, building RESTful APIs, and ensuring secure data transfer. Many libraries, such as Nodemailer, are utilized with Express.js.

3. Testing and Deployment

Manual Testing

In Manual Functional Testing, testers verify the behavior of individual functions or features of the software application by manually executing test cases. Testers interact with the software as end-users would, inputting data, triggering functions, and observing the output to ensure that the software behaves as expected.

Manual Functional Testing involves:

1. **Test Case Creation**: Testers create test cases based on requirements or specifications, outlining the steps to be followed and the expected results for each function or feature.

2. **Test Execution**: Testers manually execute the test cases, following the steps outlined in the test case documents. They input data, interact with the user interface, and perform various actions to validate the functionality of the software.

3. **Defect Reporting**: If any discrepancies or issues are found during testing, testers report them to the development team using a defect tracking system. They provide detailed information about the issue, including steps to reproduce it and any relevant screenshots or logs.

Deployment

- **Hosting platform**

For the hosting platform, we are using Hostinger.

4. Challenges Faced and solution Implemented

.

- **Bug Fixing ability:**

  I was tasked with fixing bugs such as page layout issues on mobile, images not displaying correctly, and incorrect data fetching. This improved my ability to identify and solve bugs effectively.

- **Adding New Features:**

  I learned to seamlessly integrate new features into our application, collaborating with teammates and ensuring compatibility with existing code. For instance, I added a feature to display urgent notices on the homepage.

- **Learning the Usage of Correct Tools Based on Situation:**

  I learned to identify and employ the correct tools based on the situation, ensuring optimal productivity and effective problem-solving.

- **Problem-Solving Strategies:**

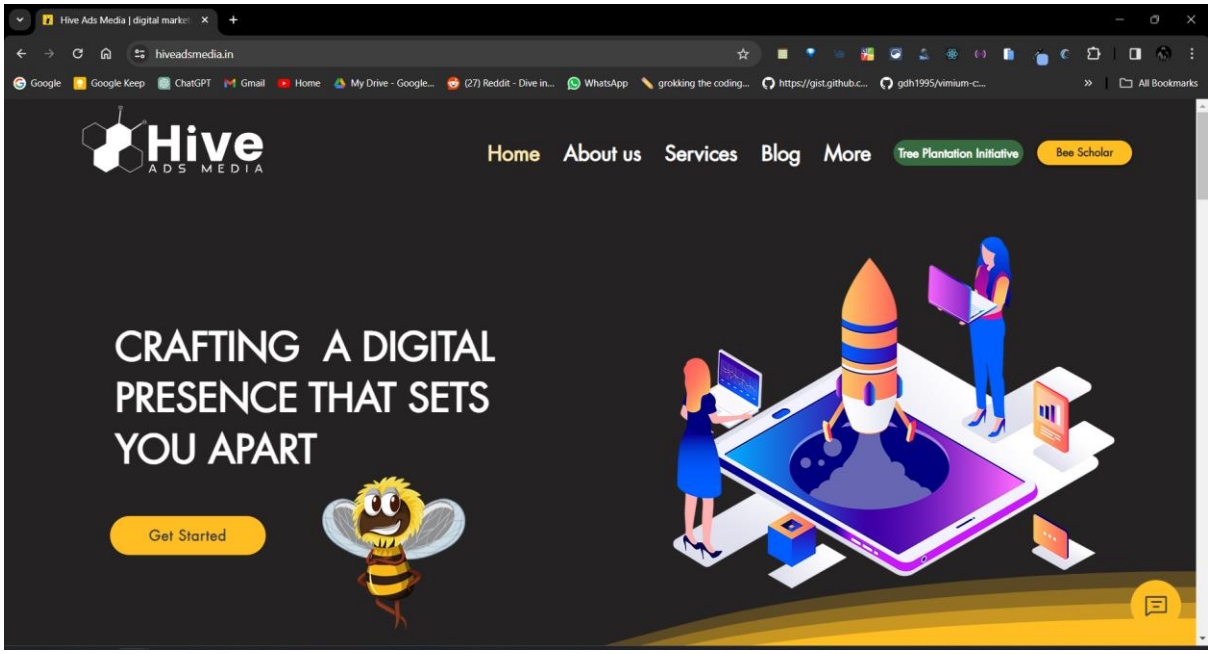
  I approached challenges with analytical thinking and creativity, systematically analysing root causes and exploring solutions to drive continuous improvement.

  I successfully completed all tasks, receiving credit from my team leader for delivering them early. This experience has enhanced my debugging and development speed and deepened my understanding of React and Node.js lifecycles.

5. Lesson Learned

I learned how to utilize different plugins to complete tasks quickly and effectively. Additionally, I gained insights into how content management systems operate behind the scenes. This knowledge empowers me to opt for WordPress as a full backend solution for simpler static websites, offering built-in editors for easy content editing. Moreover, I discovered that I can incorporate HTML and CSS to customize designs according to specific requirements.

6. Implementation



7. CONCLUSION

Developing this website enhanced my understanding of content management systems and emphasized the importance of focusing on design aspects. For static websites with minimal interactive elements like forms, WordPress proves to be a valuable tool, significantly reducing development time while enabling the creation of visually appealing websites in a short timeframe.

Continuous Engagement in Learning and Work Ethics

Learning Opportunities:

Upon reviewing the web application, I noticed that the UI was not very good. Consequently, I suggested changing the entire UI using the modern framework Tailwind CSS, identifying an opportunity to enhance the system's frontend for a more user-friendly experience. Acknowledging the importance of continuous improvement, I began learning JavaScript and jQuery, both of which were utilized in our EMS. I also acquired knowledge in WordPress and React, which were relatively new to me. This learning initiative involved extensive research, YouTube tutorials, and practical application. The acquired knowledge wasn't limited to theoretical understanding but was immediately applied to the project, resulting in significant improvements to both my skills and the EMS's user interface.

Critical Elements:

- Initiative and Proactivity: Taking the initiative to identify a skill gap and proactively addressing it demonstrated a keen sense of responsibility and eagerness to contribute to the project's success.
- Practical Application: The gained knowledge wasn't confined to theoretical understanding but was immediately applied to the project, resulting in tangible improvements in the user interface.
- Commitment to Up-to-Date Skills: By staying updated with relevant technologies like JavaScript and jQuery, I showcased a commitment to maintaining a skill set aligned with industry standards, ensuring the system remains competitive and user-friendly.

Collaboration

In this collaborative project sprint, effective communication and collaboration with my team were critical features that emerged as vital components of success. Regular team meetings, constant chat communication during coding, idea exchange, updates, and the use of Git for collaboration were integral to the process.

Critical Elements:

- Communication Channels: Regular team meetings fostered an environment where team members felt comfortable expressing ideas, raising concerns, and sharing progress updates.
- Transparency and Accountability: Project management tools like Trello introduced transparency into task allocation and progress tracking, holding team members accountable for their respective responsibilities.

- Streamlined Workflow: The combination of effective communication and project management tools streamlined the workflow, minimizing potential bottlenecks and contributing to the overall success of the project.

Problem Solving:

Encountering a problem while integrating the Tailwind library provided an opportunity to showcase my problem-solving skills. I also demonstrated my ability to track down a challenging bug - our system, responsible for live-tracking employee tasks, kept logging out repeatedly. We implemented a solution using a WebSocket instead of relying solely on HTTPS for live time tracking. Additionally, I improved the performance of the company website by redesigning it. I also fix many bugs in React web application like image not loading, mobile breakout and many other thing.

Critical Elements:

- Analytical Approach: The use of profiling tools showcased an analytical approach to problem-solving, identifying specific areas impacting system performance. Also, the UI's lack of interactivity was addressed.
- Collaborative Problem-Solving: Addressing a performance bottleneck often requires collaboration. I worked closely with team members to implement optimization techniques, drawing on collective expertise to achieve the best results for the company website and resolving the bug through the use of WebSocket.
- User-Centric Focus: The ultimate goal of problem-solving was to enhance the user experience. This approach ensured that technical solutions aligned with the overarching objective of delivering a high-performing and user-friendly enterprise management system. I suggested using Tailwind CSS for this purpose.

Conclusion

The development of the Enterprise Management System and the web application using React.js, Express.js, and Node.js exemplify how diligent work, adherence to engineering standards, and a commitment to quality contribute to project success. In both cases, the outcome was a robust, scalable, and secure solution that met organizational requirements while adhering to industry best practices. These projects provided valuable learning opportunities, allowing for the improvement of programming skills, communication, collaboration, and the articulation of ideas.

Additionally, the experience gained from developing these projects underscored the significance of content management systems (CMS) in web development. While the Enterprise Management System showcased the importance of a custom-built solution tailored to specific organizational needs, the use of WordPress for simpler static websites highlighted the efficiency and ease of use provided by CMS platforms. By leveraging WordPress, development time was significantly reduced without compromising on the quality or aesthetics of the final product.

Overall, these projects underscore the importance of adaptability and utilizing the right tools for the task at hand. While custom solutions may be necessary for complex applications, platforms like WordPress offer a valuable resource for rapid development and efficient deployment of static websites. This experience has reinforced the importance of continuous learning and adapting to new technologies and methodologies in the ever-evolving landscape of web development.

Signature (Student)

Signature (Guide)