CHAPTER 1 - OVERVIEW OF THE COMPANY

1.1 HISTORY

RKIT Software Pvt. Ltd. was started in 1998 by three young entrepreneurs. Today we are amongst the top 10 accounting software providers in India.

Initially, the company started the development of customized software and gradually shifted to different kinds of Packaged Accounting Software. General-purpose accounting software developed by the company is very popular in the market and brand leader in their respective areas with more than 75,000 users.

Company's motto in the development of software packages is 'Keep It Easy & Advance'. Users feel comfortable while using our products and do not require any special skills to use it.

Our main product is 'Miracle'. The miracle is general-purpose accounting software that is scalable from the accountant to the small shop owner to big industries. My Investor, Miracle-G, Miracle-Mandi, and Miracle-Petro are other products developed by the company.

RKIT Software Pvt. Ltd. is based at Rajkot (Gujarat) and has branch offices at Ahmedabad, Vadodara, Mehsana in Gujarat and Nagpur in Maharashtra. The company has a dealer network in Gujarat, Maharashtra, Madhya Pradesh, Chhattisgarh, Rajasthan, Assam and Tamil Nadu and is expanding to other states of India soon. The company has a dealer network in Oman and is expanding to other Arab Countries.

1.2 SCOPE OF WORK

Our company is currently working on several products, including My Investor, Miracle-G, Miracle-Mandi, and Miracle-Petro, as well as other products in the pipeline. The scope of work and details of each product are outlined below:

1. **My Investor**: My Investor is a comprehensive investment management software that offers a user-friendly platform for investors to manage their portfolio, track investments, analyze performance, and make informed investment decisions. The scope of work includes designing and developing a robust and scalable web-based application with advanced features such as real-time portfolio tracking, investment performance analytics, risk assessment, goal-based investing, and integration with financial data sources.
2. **Miracle-G**: Miracle-G is a mobile app for farmers that provides real-time weather updates, crop advisory, and market prices to help farmers make informed decisions about their farming practices. The scope of work includes designing and developing a user-friendly mobile application that integrates with various data sources to provide accurate weather information, crop-specific advisory, market prices, and other relevant information to farmers in their local language.
3. **Miracle-Mandi**: Miracle-Mandi is an e-commerce platform that connects farmers with buyers, processors, and exporters, enabling them to sell their produce at fair prices and gain access to a wider market. The scope of work includes developing a secure and scalable online marketplace that facilitates seamless transactions, supports multiple payment options, and includes features such as product listings, bidding, order management, and logistics integration.
4. **Miracle-Petro**: Miracle-Petro is a specialized software solution for the oil and gas industry that helps in optimizing operations, managing assets, monitoring production, and analysing data for improved decision-making. The scope of work includes developing a comprehensive software solution that incorporates data analytics, machine learning, and visualization tools to streamline operations, track asset performance, forecast production, and optimize resource allocation.

In addition to the above products, our company is also working on other innovative products in the pipeline, which are aimed at addressing various industry-specific challenges and providing value-added solutions to our clients. The scope of work for these products will be defined based on their individual requirements and goals.

Overall, the scope of work for our products includes conceptualization, design, development, testing, deployment, and ongoing support to ensure that our clients receive high-quality, reliable, and scalable software solutions that meet their specific needs and drive their business growth.

CHAPTER 2 - OVERVIEW OF DIFFERENT DEPARTMENT AND LAYOUT

2.1 DETAILS OF WORK BEING CARRIED OUT IN THE DEPARTMENT

1. **HR Department**:

   The HR department is responsible for managing various aspects of human resources, including recruitment, onboarding, employee engagement, performance management, training and development, compensation and benefits, employee relations, and compliance with labour laws and regulations. The HR team works closely with other departments to ensure that the company has a skilled and motivated workforce and promotes a positive work culture.

2. **Marketing Department**:

   The Marketing department is responsible for creating and implementing marketing strategies to promote the company's products or services, build brand awareness, generate leads, and increase customer acquisition and retention. The team conducts market research, develops marketing campaigns, manages social media and online presence, creates content, designs and executes marketing collateral, organizes events, and analyzes marketing performance to drive business growth.

3. **Development Department**:

   The Development department is responsible for designing, developing, and maintaining software products or solutions offered by the company. This includes coding, testing, debugging, and documentation of software applications, databases, websites, or other software assets. The team works on front-end and back-end technologies, follows best practices, ensures software quality and security, and collaborates with other departments to understand client requirements and deliver solutions that meet customer needs.

4. **Customer Support Department**:

   The Customer Support department is responsible for providing assistance and resolving technical issues for customers who are using the company's products or services. This includes responding to customer inquiries, diagnosing and resolving technical problems, providing product information and training, managing customer complaints, and maintaining customer satisfaction. The team works closely with the Development department to escalate and resolve complex technical issues, and ensures timely and effective customer support.

5. **Research Team**:

   The Research team is responsible for conducting market research, competitor analysis, and technology trends analysis to identify new business opportunities, explore potential markets, and stay updated with industry developments. The team collects and analyses data, conducts experiments or studies, generates insights and recommendations, and collaborates with other departments to align business strategies with market insights.

Each department plays a crucial role in the overall functioning of the company, and their work is aligned towards achieving the company's goals, meeting customer needs, and driving business success. Collaboration and coordination among departments are key to ensuring smooth operations and delivering high-quality products or services to customers.

2.2 TECHNICAL SPECIFICATIONS

Company specializes in front-end technologies such as HTML, CSS, JavaScript, jQuery, and AJAX for building interactive and visually appealing user interfaces for web applications. Also used DevExtream jQuery library for GUI application. These technologies are widely used in modern web development to create responsive and user-friendly web application. Company team of front-end developers ensures that the software developed are visually appealing, accessible, and optimized for performance across different devices and browsers. This involves creating responsive layouts, implementing smooth animations, and utilizing AJAX for seamless data retrieval and updates without page reloads. These front-end technologies provide a solid foundation for delivering engaging and user-centric web applications.

On the back-end, Company utilizes the .NET Framework and .NET Core WEB API for building scalable and high-performance web APIs. These technologies allow for building robust and secure server-side logic for web applications, handling data processing, authentication, and business logic. Company team of back-end developers ensures that the APIs are optimized for performance, maintainability, and security. Additionally, Company uses MySQL as the database management system for storing, retrieving, and managing data efficiently. Along with web development, Company also specializes in mobile app development using Flutter for cross-platform app development and Android for native app development. This includes designing and building mobile apps with rich user interfaces, smooth navigation, and seamless integration with APIs and databases.

Overall, Company's technical specifications include expertise in front-end technologies such as HTML, CSS, JavaScript, jQuery, and AJAX, back-end technologies such as .NET Framework and .NET Core WEB API, MySQL as the database management system, and mobile app development using Flutter and Android. These technologies and tools enable your company to deliver modern, scalable, and high-performance web and mobile applications for a diverse range of clients and industries.

2.3 Sequence of operation for development of product

1. **Requirements Gathering**: The first step in the development process is gathering requirements from stakeholders, including customers, end users, and other relevant parties. This involves identifying the desired features, functionalities, and performance expectations of the end product.
2. **Analysis and Design**: Once the requirements are gathered, the next step is to analyze and design the software system. This involves creating detailed specifications, defining the system architecture, designing the user interface, and creating data models and flowcharts to map out the system's structure and behaviour.
3. **Development**: After the analysis and design phase, the development team starts coding the software according to the specifications and design documents. This involves writing the actual code using programming languages, frameworks, and tools as appropriate, and integrating different software components to create the end product.
4. **Testing**: Once the software is developed, it undergoes various testing activities to identify and fix defects, errors, and vulnerabilities. This includes unit testing, integration testing, system testing, and other types of testing depending on the company's testing strategy. The goal is to ensure that the software meets the requirements and functions correctly.
5. **Deployment**: Once the software is tested and approved, it is deployed to the production environment or released to the customers. This involves installing the software on the target hardware, configuring it, and making it available for end users to access and use.
6. **Maintenance** and Support: After deployment, the software may require ongoing maintenance and support to ensure its smooth operation. This may include fixing bugs, addressing issues, providing updates, and providing technical support to end users.
7. **Monitoring and Optimization**: Once the software is deployed and in use, it's important to monitor its performance and optimize its functionality. This may involve gathering usage data, analysing performance metrics, and making improvements to enhance the software's performance and user experience.

8. **Continuous Improvement**: Throughout the entire development process, feedback from stakeholders and end users is collected and used to continuously improve the software. This may involve incorporating new features, addressing user feedback, and making updates based on changing requirements or technologies.

2.4 DETAILS ABOUT EACH STAGE OF PRODUCTION.

1 Agile Planning:

Product Backlog: The Agile team, including the Product Owner, creates and maintains a prioritized list of features, functionalities, and user stories, known as the product backlog. This backlog is continuously refined and updated throughout the development process based on feedback and changing requirements.

Sprint Planning: Before each sprint, the team conducts a sprint planning meeting, where they select a set of backlog items to work on during the sprint. The Development Team estimates the effort required for each item and collaboratively decides how much work can be taken on based on their capacity and velocity.

Definition of Done: The team establishes a clear definition of done, which outlines the criteria that must be met for a backlog item to be considered complete. This definition includes coding, testing, documentation, and other quality standards.

2 Sprint Execution:

Design: The Development Team designs the software based on the selected backlog items, using best practices and established design patterns. This may include creating wireframes, mockups, or prototypes to visualize the user interface and overall system architecture.

Coding: The Development Team writes the code for the software, following coding standards and guidelines. They use version control systems to manage code changes and collaborate on code development.

Testing: The Development Team writes automated tests and conducts various types of testing, including unit testing, integration testing, and acceptance testing, to ensure that the software meets the defined criteria of done and is free from defects.

Collaboration: The team collaborates closely with the Product Owner, stakeholders, and other team members, including testers, to clarify requirements, resolve issues, and ensure that the software is being developed according to the intended functionality.

1. **Daily Stand-up Meetings:**

   Daily Stand-up: The team holds short daily stand-up meetings, usually lasting 15 minutes, where they share updates on progress, discuss challenges, and plan for the day. Each team member answers three questions: What did I do yesterday? What am I planning to do today? Are there any obstacles or roadblocks?

2. **Sprint Review:**

   Demo: At the end of each sprint, the team conducts a sprint review, where they demonstrate the completed work to the Product Owner, stakeholders, and other relevant parties. This allows for feedback and validation of the product, and helps to ensure that it meets the intended requirements.

   Feedback: The Product Owner and stakeholders provide feedback on the demonstrated work, which may include suggestions for improvements, clarifications on requirements, or identifying any gaps or issues. This feedback is taken into account for refining the product backlog and planning future sprints.

3. **Sprint Retrospective:**

   Reflect: After the sprint review, the team conducts a sprint retrospective, where they reflect on the sprint and discuss what went well, what could have been done better, and what actions can be taken to improve the development process. This allows the team to continuously learn and make adjustments for future sprints.

   Action Items: Based on the sprint retrospective, the team identifies specific actions to be taken in the next sprint to improve the development process. These actions may include process improvements, technical enhancements, or communication adjustments, and are documented as action items for implementation in the next sprint.

4. **Backlog Refinement:**

   Continuous Refinement: Throughout the sprint and development process, the team continuously refines the product backlog. This includes adding, removing, or reprioritizing backlog items based on feedback, changing requirements, or market demands.

   Detailed Requirements: The Product Owner works with stakeholders and the Development Team to further refine and clarify the requirements for backlog items. This may include creating detailed user stories, defining acceptance criteria, and documenting any dependencies or constraints.

5. **Continuous Integration and Testing:**

   Continuous Integration:

CHAPTER 3 - INTERNSHIP MANAGEMENT

3.1 INTERNSHIP SUMMARY

During my internship, I gained valuable experience in various front-end and back-end technologies, acquiring core knowledge in all the technologies used by the company. I was actively involved in performing several tasks assigned by the company, and was given the mini project called "Employee Management System" which covered all the topics I learned during my internship training.

Throughout my internship, I had the opportunity to work with different technologies, including front-end technologies such as HTML, CSS, and JavaScript, as well as back-end technologies like .NET Framework for web API. I also gained hands-on experience in working with databases such as MySQL for data storage and retrieval.

One of the highlights of my internship was working on the "Employee Management System" mini project. This project allowed me to apply my knowledge of front-end and back-end technologies to create a fully functional system for managing employee information, including features such as employee registration, login, profile management, and data retrieval. I was responsible for designing and implementing the user interface, developing the back-end logic for handling user requests, and integrating the front-end and back-end components to create a seamless user experience.

During my internship, I also had the opportunity to collaborate with other team members, including senior developers and project managers, to understand the project requirements, gather feedback, and make necessary adjustments to improve the functionality and performance of the system. I also participated in regular team meetings and discussions, which helped me gain insights into the software development process and learn about best practices in the industry.

Overall, my internship experience provided me with valuable hands-on experience in working with front-end and back-end technologies, as well as exposure to real-world project development in a professional setting. I developed a solid understanding of the software development lifecycle, learned how to collaborate with a team, and gained practical skills that I can apply in my future career as a full stack developer.

3.2 PURPOSE

The purpose of an internship as a full stack developer is to provide aspiring developers with the opportunity to gain practical experience and develop skills in both front-end and back-end technologies. It allows interns to apply the concepts and principles of full stack development in real-world projects, understand the complete web development process, and enhance their teamwork, project management, and problem-solving abilities. Interns can work on diverse projects, ranging from designing user interfaces to developing server-side logic, handling databases, implementing authentication and authorization, and integrating front-end and back-end components to create complete web applications. By working on practical projects during the internship, interns can build a strong foundation in full stack development and gain valuable industry experience.

As part of the internship, interns are often given the opportunity to create a mini project, such as an Employee Management System. The mini project may involve designing and implementing user interfaces, developing server-side logic to handle employee data, implementing authentication and authorization for secure access, and integrating front-end and back-end components for seamless functionality. It provides me with a hands-on experience of building a real-world web application, and serves as a valuable addition to their portfolio, showcasing their practical skills and expertise in full stack development.

3.3 OBJECTIVE

The objectives of creating a mini project as part of my internship, based on learning front-end and back-end technologies, were:

Apply Front-End Technologies: The mini project aimed to allow me to apply the front-end technologies such as HTML, CSS, and JavaScript that I learned during my internship training. It provided an opportunity to create user interfaces that are visually appealing, responsive, and user-friendly, following best practices in front-end development.

Implement Back-End Functionality: The mini project aimed to enable me to implement the back-end functionality of the web application using back-end technologies such as .NET Framework, and databases. This involved handling user requests, processing data, implementing authentication and authorization, and implementing server-side logic to provide desired functionality.

Integrate Front-End and Back-End Components: The mini project aimed to allow me to integrate the front-end and back-end components of the web application to create a seamless user experience. This involved designing APIs, implementing RESTful services, and establishing communication between the front-end and back-end components to ensure smooth data flow and interactions between the user interface and the back-end logic.

Practice Project Management and Collaboration: The mini project aimed to provide an opportunity for me to practice project management skills and collaborate with team members. This included understanding project requirements, planning and organizing tasks, coordinating with team members, and meeting project deadlines. It helped me understand the importance of effective communication, teamwork, and coordination in a real-world software development environment.

Develop Problem-Solving and Troubleshooting Skills: The mini project aimed to allow me to develop problem-solving and troubleshooting skills by addressing challenges and issues that may arise during the project. This included debugging code, resolving technical issues, and optimizing performance to ensure a smooth functioning of the application.

Showcase Skills and Create Portfolio: The mini project aimed to provide me with a tangible outcome that I can showcase in my portfolio to demonstrate my skills and expertise in front-end and back-end technologies. It served as a valuable addition to my resume, showcasing my ability to apply theoretical knowledge to practical projects.

Overall, the objectives of creating the mini project were to apply and demonstrate my learning in front-end and back-end technologies, gain practical experience in real-world project development, develop teamwork and project management skills, and create a portfolio-worthy project to showcase my skills to potential employers.

3.4 SCOPE

The scope of an Employee Management System mini project can vary depending on the specific requirements and objectives set by the given task. However, some common scope elements for an Employee Management System mini project could include:

1. **Employee Information Management**: The mini project may involve creating a system to manage employee information, such as storing and retrieving employee details like name, contact information, job title, department, and other relevant data.
2. **User Authentication and Authorization**: The scope may include implementing user authentication and authorization mechanisms to ensure secure access to the system and manage user roles and permissions.
3. **Employee Self-Service**: The scope may include developing a self-service portal for employees to access and update their personal information, view own task given by project co-ordinator and perform actions on task.
4. **Task Creation and Assignment**: The project may involve creating a system for project co-ordinator to create tasks, assign tasks to themselves or others, and set deadlines and priorities for tasks.
5. **Task Tracking and Progress Monitoring**: The scope may include developing functionality to track the progress of tasks, update task statuses.

3.5 TECHNOLOGY AND LITERATURE REVIEW

3.5.1 .NET Framework:

The .NET Framework is a software development platform developed by Microsoft that is widely used for creating back-end web APIs (Application Programming Interfaces). It provides a comprehensive set of tools, libraries, and runtime environments for building robust, scalable, and high-performance web applications.

One of the key features of the .NET Framework is its support for creating web APIs using ASP.NET, a popular web application framework. ASP.NET allows developers to build RESTful web APIs that can be used to expose data and services over the internet, which can be consumed by clients such as web browsers, mobile apps, and other servers.

The .NET Framework also includes a vast set of libraries and APIs for handling various aspects of back-end web development, such as data access, security, authentication, caching, and error handling. It supports multiple programming languages, including C#, Visual Basic, and F#, and provides a rich set of tools and IDEs (Integrated Development Environments) like Visual Studio for building and debugging web APIs efficiently.

Furthermore, the .NET Framework also offers extensive support for scalability, performance, and security, making it a preferred choice for building enterprise-grade web APIs. It includes features like caching, performance optimizations, authentication and authorization, encryption, and exception handling, which are crucial for creating robust and secure web APIs.

In summary, the .NET Framework is a powerful and versatile software development platform used for building back-end web APIs. It provides a rich set of tools, libraries, and runtime environments for creating scalable, high-performance, and secure web applications. ASP.NET, the web application framework within the .NET Framework, offers a wide range of features for building RESTful web APIs, making it a popular choice among developers for creating robust and reliable back-end solutions.

3.5.2 Front End Technologies:

1. **HTML (Hypertext Markup Language)**: HTML is the standard markup language used to create the structure and content of web pages. It provides a set of predefined tags that define elements such as headings, paragraphs, images, links, forms, and more.
2. **CSS (Cascading Style Sheets)**: CSS is a stylesheet language used to control the visual appearance of web pages. It allows developers to define the style and layout of HTML elements, such as fonts, colors, spacing, borders, and more, making it possible to create visually appealing and responsive web designs.
3. **Bootstrap**: Bootstrap is a popular front-end framework that provides a collection of pre-designed CSS and JavaScript components for building responsive and mobile-first web pages. It includes a grid system, UI components like buttons, forms, and navigation, and JavaScript plugins for common tasks like modal dialogs, carousels, and more.
4. **JavaScript**: JavaScript is a widely used programming language that enables web developers to add interactivity and dynamic content to web pages. It can be used to manipulate HTML elements, handle user events, perform form validations, make asynchronous requests to servers, and more, allowing for dynamic and interactive user experiences on the web.
5. **jQuery**: jQuery is a fast, lightweight, and popular JavaScript library that simplifies DOM (Document Object Model) manipulation, event handling, and AJAX interactions. It provides a set of easy-to-use functions and methods that make it easier for developers to perform common tasks, such as traversing and manipulating HTML elements, handling events, making AJAX requests, and animating elements.
6. **AJAX (Asynchronous JavaScript and XML)**: AJAX is a technique used to create asynchronous web applications that can update parts of a web page without requiring a full page reload. It combines JavaScript, XML (or other data formats, such as JSON), and HTTP requests to allow web pages to dynamically update content and interact with servers without interrupting the user's experience.

3.5.3 MySQL:

MySQL is a popular open-source relational database management system (RDBMS) that is widely used for building web applications and other software solutions. It is known for its ease of use, flexibility, scalability, and reliability, making it a popular choice among developers and businesses.

MySQL allows developers to store, manage, and retrieve data in a structured manner using tables with rows and columns. It supports various data types, including numeric, string, date and time, and more, allowing for versatile data storage and retrieval. MySQL also provides powerful features such as indexing, transactions, and stored procedures, which enhance its performance and security.

One of the key strengths of MySQL is its wide support for programming languages, frameworks, and platforms, making it compatible with a broad range of applications and development environments. It has official connectors and APIs for popular programming languages like PHP, Java, Python, .NET, and more, making it easy to integrate with different programming languages and frameworks.

MySQL is also known for its scalability, allowing developers to handle large volumes of data and concurrent users efficiently. It supports features such as replication and clustering, which enable high availability and load balancing for mission-critical applications.

In addition, MySQL has a large and active community of developers, users, and contributors, providing extensive documentation, tutorials, forums, and support. This makes it easier for developers to learn, troubleshoot, and optimize their MySQL databases.

3.6 INTERNSHIP PLANNING

3.6.1 INTERNSHIP DEVELOPMENT APPROACH AND JUSTIFICATION

As an intern, my development approach involves actively learning from experienced professionals, contributing to real-world projects, and gaining practical experience in a professional work environment. I joined the company as an intern to apply and reinforce the skills I acquired during my academic or training programs, learn from real-world projects, and establish valuable connections in the industry. Through my internship, I aim to gain hands-on experience, develop professional skills, and explore potential career paths, making it a strategic step towards my professional growth and development.

3.6.2 INTERNSHIP EFFORT AND TIME, COST ESTIMATION

Month 1:

- Focus on learning GUI technologies such as HTML, CSS, JavaScript, jQuery, and AJAX.
- Study and understand the concepts of front-end web development.
- Practice with hands-on exercises to reinforce the learning.
- Build small demos to apply the learned concepts and gain practical experience.

Month 2:

- Shift focus to basic C# programming language and Web API.
- Learn the fundamentals of C# programming language, including syntax, data types, and basic programming concepts.
- Understand the basics of building Web APIs using C# and learn about RESTful API principles.
- Practice with hands-on exercises to reinforce the concepts and build simple Web APIs.

Month 3:

- Advance learning in C# programming language and Web API.
- Dive deeper into advanced topics in C# programming language, including object-oriented programming, exception handling, and LINQ.
- Gain a solid understanding of Web API concepts such as authentication, routing, and data validation.
- Apply the learned concepts from the first two months to a real-world project.
- Create a comprehensive Employee Management System that covers front-end technologies (HTML, CSS, JavaScript, jQuery, AJAX), back-end development with C# and Web API, and database integration.
- Work on the project, applying best practices and industry standards in software development.
- Collaborate with team members, receive feedback, and make necessary improvements to the project.

- Gain practical experience in building a full-stack web application, from front-end to back-end, and complete the project as a showcase of the acquired skills.

Throughout the internship, the emphasis would be on hands-on learning, practical application of concepts, and gaining real-world experience in web development using front-end technologies, C#, and Web API. The learning would be progressive, building on the knowledge acquired in the previous months, and culminating in a project that covers all the topics learned during the internship.

3.6.3 ROLES AND RESPONSIBILITIES

As an intern in a software company, my roles and responsibilities revolved around front-end and back-end development using various technologies. Here is a summary of my roles and responsibilities:

Front-end Development:

- Learning, Creating and designing user interfaces using HTML, CSS, and Bootstrap to develop visually appealing and responsive web pages.
- Implementing JavaScript, jQuery, and AJAX to enhance user interactions and improve the overall user experience.
- Collaborating with UX/UI designers and other team members to ensure consistent design and user experience across different devices and browsers.
- Testing and debugging front-end code to ensure smooth functionality and optimal performance.
- Keeping up-to-date with the latest front-end development trends and best practices to continuously improve the quality of the front-end code.

Back-end Development:

- Utilizing .NET Framework and C# to develop Web APIs for data retrieval, processing, and storage.
- Implementing database operations using MySQL to store and retrieve data efficiently.
- Collaborating with the front-end team to integrate front-end interfaces with back-end APIs for seamless data flow.
- Ensuring the security and integrity of data by implementing appropriate authentication and authorization mechanisms.
- Troubleshooting and resolving issues related to back-end functionality, performance, and scalability.

Overall, my roles and responsibilities as an intern involved developing and maintaining front-end interfaces using HTML, CSS, Bootstrap, JavaScript, jQuery, and AJAX, and back-end development using .NET Framework, C#, and MySQL. I worked closely with other team members, followed best practices, and stayed updated with the latest industry trends to deliver high-quality web applications that meet the company's requirements and standards.

3.6.4 GROUP DEPENDENCIES

As an individual working on a mini project, I was responsible for both the front-end and back-end development. The front-end dependencies included HTML, CSS, Bootstrap, JavaScript, and jQuery, which were used for building the user interface, styling, and interactivity of the web application. On the back-end, I worked with the .NET Framework, C#, Web API, and MySQL database for server-side logic, data retrieval, processing, and storage. These technologies were integrated to create a cohesive and functional web application. As an individual, I had to manage and coordinate the dependencies between the front-end and back-end technologies to ensure smooth integration and successful completion of the mini project.

3.7 INTERNSHIP SCHEDULING

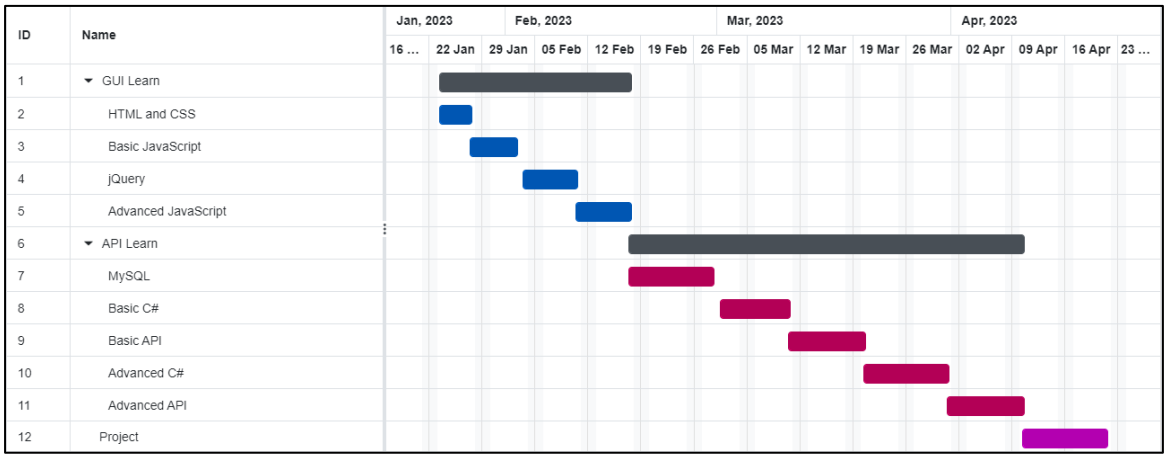

Fig 3.1 (Gantt Chart)

CHAPTER 4 - SYSTEM ANALYSIS

4.1 STUDY OF CURRENT SYSTEM

The study of the current system of a web application reveals that it is capable of managing employee personal details and providing administrative functionalities. The system allows the administrator to manage both their own details as well as those of employees. It provides features such as adding, editing, and deleting employee records, updating personal information, and managing tasks assigned to specific employees. The administrator can also keep track of tasks and maintain records of all tasks in a centralized manner. The system allows for efficient management of employee details and tasks, streamlining the administrative processes and enhancing productivity. However, further analysis may be required to identify any potential areas of improvement or additional features that could enhance the system's performance and meet the needs of the organization more effectively.

4.2 PROBLEM AND WEAKNESSES OF CURRENT SYSTEM

Lack of Automation: The current system may rely heavily on manual data entry and task assignment, leading to increased chances of errors and inefficiencies. For example, tasks may not be automatically assigned based on priorities or workload, resulting in uneven distribution of workloads among employees.

Limited Scalability: The current system may not be designed to handle a large number of employees or tasks, which could pose challenges as the organization grows or as the workload increases. This could result in performance issues, slow response times, or system crashes, impacting overall productivity.

Lack of Advanced Reporting and Analytics: The current system may not have robust reporting and analytics capabilities, making it challenging for the administrator to generate insights on employee performance, task progress, or workload distribution. This could hinder effective decision-making and strategic planning.

Security and Privacy Concerns: The current system may not have adequate security measures in place to protect employee data and other sensitive information. This could expose the organization to data breaches, unauthorized access, or privacy violations, posing potential legal and reputational risks.

Limited Accessibility and User Experience: The current system may have limitations in terms of accessibility and user experience. For example, it may not be mobile-friendly, making it difficult for employees or the administrator to access and manage tasks on the go. This could result in lower user adoption and reduced overall effectiveness.

Lack of Integration with Other Systems: The current system may not be integrated with other relevant systems or tools used in the organization, such as HR management systems or project management tools. This could result in data silos and duplication of efforts, leading to inefficiencies and inconsistencies in data management.

Lack of Customization: The current system may not be easily customizable to meet the specific needs and requirements of the organization. This could limit the system's flexibility and adaptability to changing business processes or unique workflows.

4.3 SYSTEM FEASIBILITY

1. **Technical Feasibility**: This assesses the technical capabilities of the system, including the availability of necessary hardware, software, and technical expertise to develop and maintain the system. It evaluates whether the required technology is readily available, reliable, and compatible with existing systems and infrastructure.
2. **Economic Feasibility**: This evaluates the economic viability of the system, including the cost of development, implementation, and maintenance in comparison to the potential benefits and return on investment. It involves analyzing the costs and benefits associated with the system, including hardware, software, training, and ongoing operational costs, and determining whether the system is financially feasible for the organization.
3. **Operational Feasibility**: This assesses the operational viability of the system, including its compatibility with existing business processes, policies, and procedures. It involves analyzing the impact of the system on day-to-day operations, user acceptance, and the ability to integrate with other systems and processes.
4. **Schedule Feasibility**: This evaluates the time frame required to develop, implement, and deploy the system, and whether it aligns with the organization's timelines and deadlines. It involves assessing the feasibility of meeting project milestones, deadlines, and delivery schedules.

4.4 ACTIVITY / PROCESS IN NEW SYSTEM

The new system of **Employee Management System** would involve several activities and processes to effectively manage employee details and tasks. Some of the key activities or processes in the system may include:

1. **Employee Registration**: The system would allow for employee registration, including capturing and storing employee personal details such as name, contact information, job title, department, and other relevant information.
2. **Employee Profile Management**: The system would enable employees and administrators to manage their profiles by updating personal information, uploading profile pictures, and maintaining accurate and up-to-date records.
3. **Task Assignment**: The system would allow administrators to assign tasks to specific employees, including defining task details, deadlines, and priorities. Employees would be notified of the assigned tasks and can view and access them from their dashboard.
4. **Task Tracking**: The system would provide a tracking mechanism for employees to update the status of assigned tasks, such as in-progress, completed, or pending. Administrators can also track the progress of tasks, generate reports, and monitor overall task performance.
5. **Task Reminders and Notifications**: The system would send reminders and notifications to employees and administrators for upcoming deadlines, task updates, and other relevant information to ensure timely task completion.
6. **Employee Leave Management**: The system may have a module to manage employee leave requests, including submitting leave applications, tracking leave balances, and managing leave approvals.
7. **System Administration**: The system would have administrative features to manage user roles, permissions, system settings, and other administrative tasks to ensure proper system configuration and security.

These are some of the key activities and processes that may be included in a new system of employee and task management. The exact features and functionalities would depend on the specific requirements and scope of the system as defined by the organization. It is important to carefully design and develop the system to meet the needs of the organization and its employees, and provide an efficient and user-friendly solution for managing employee details and tasks.

4.5 FEATURES OF NEW SYSTEM

The new system of employee and task management would aim to provide a comprehensive and efficient solution for managing employee details and tasks. Some of the key features that may be included in the system could include:

1. Employee Registration and Profile Management
2. Task Assignment and Tracking
3. Task Reminders and Notifications
4. Mobile Accessibility
5. User Roles and Permissions
6. Leave Management

4.6 List of Main Modules

1. Admin Management Panel
2. Employee Management Panel
3. Task Management Panel

4.7 SELECTION OF HARDWARE / SOFTWARE / TECHNOLOGIES / APPROACHES AND JUSTIFICATIONS

Hardware:

Computer: A modern desktop or laptop computer with sufficient processing power, memory, and storage capacity would be suitable for running the employee and task management system.

Software:

- **Visual Studio**: Visual Studio is a widely used integrated development environment (IDE) for developing web applications in C# and .NET, providing robust tools and features for development, debugging, and testing.
- **Chrome**: Chrome is a popular web browser with excellent developer tools for testing and verifying web applications.
- **Visual Studio Code**: Visual Studio Code is a lightweight, free, and open-source code editor with a rich set of extensions for web development.
- **Workbench**: Workbench is a graphical user interface tool for managing MySQL databases, providing a convenient way to design, develop, and manage database schemas, tables, and queries.

Technologies:

C# and .NET: C# and .NET are widely used technologies for building web applications in the Microsoft ecosystem, providing a robust and scalable framework for development.

MySQL: MySQL is a popular open-source relational database management system that can be used for managing the database of the employee and task management system.

Approaches:

Client-Server Architecture: The client-server architecture allows for clear separation of responsibilities between the client-side and server-side components, enabling scalability, maintainability, and flexibility in the system's design.

Web-based: A web-based approach allows for easy access and usage through web browsers without requiring additional software installation, providing remote access, cross-platform compatibility, and scalability.

Justifications:

- Visual Studio, Visual Studio Code, and Workbench are widely used and popular tools in the software development industry, providing robust features, excellent support, and a large developer community.
- Chrome is known for its performance, compatibility, and developer-friendly features, making it ideal for testing and verifying web applications.
- C# and .NET are suitable for web application development in the Microsoft ecosystem, providing a robust and scalable framework.
- MySQL is a popular and widely used database management system, suitable for managing the database of the employee and task management system.
- The client-server architecture allows for separation of concerns, scalability, and flexibility in the system's design and development.
- The web-based approach provides easy access, cross-platform compatibility, and scalability, making it convenient for employees and administrators to access and use the system from various devices.

CHAPTER 5 - SYSTEM DESIGN

5.1 SYSTEM DESIGN & METHODOLOGY

The system design for the Employee Management System involves several key components, including the system architecture, database design, user interface design, and system flow. The system is designed to be user-friendly, efficient, and scalable, with a focus on meeting the requirements of managing employee details and tasks effectively.

System Architecture:

The system follows a client-server architecture, where the client-side is the user interface accessible through web browsers, and the server-side is responsible for processing requests, managing the database, and performing business logic. The client-side and server-side communicate through APIs (Application Programming Interfaces) for exchanging data and functionalities.

Database Design:

The database is designed to store employee details, task information, and other relevant data. It includes tables such as Employee, Task, and User, with appropriate relationships and attributes to store the necessary information. The database is designed to be normalized, ensuring data integrity, and minimizing redundancy.

User Interface Design:

The user interface is designed to be intuitive and easy to use. It includes screens for managing employee details, task assignments, task tracking, and other functionalities. The user interface is designed using modern web technologies such as HTML5, CSS3, and JavaScript, with responsive design to ensure compatibility across different devices and screen sizes.

System Flow:

The system follows a logical flow of activities, starting with user authentication and authorization, followed by managing employee details, assigning tasks to employees, tracking task progress, and generating reports. The system flow is designed to be efficient and streamlined, ensuring that users can easily navigate through different functionalities and perform their tasks effectively.

Methodology:

The system development follows an Agile methodology, which involves iterative development, continuous feedback, and regular updates based on user requirements. The Agile methodology allows for flexibility in incorporating changes and enhancements based on user feedback and evolving business needs. The development process includes stages such as requirements gathering, system design, implementation, testing, and deployment, with regular reviews and evaluations at each stage.

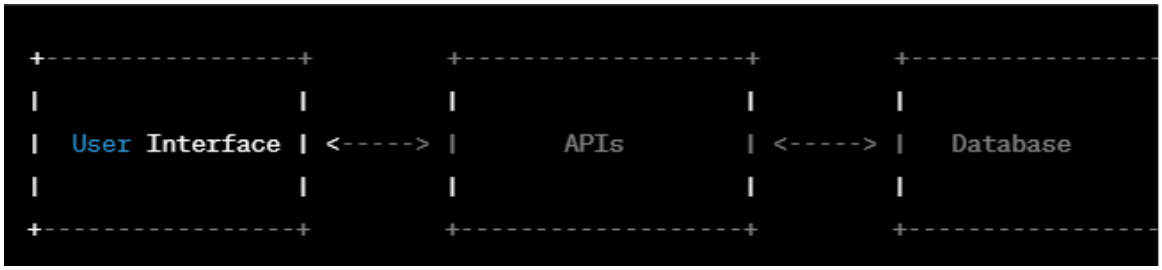

Fig. 5.1 (System Design)

5.2 DATABASE DESIGN

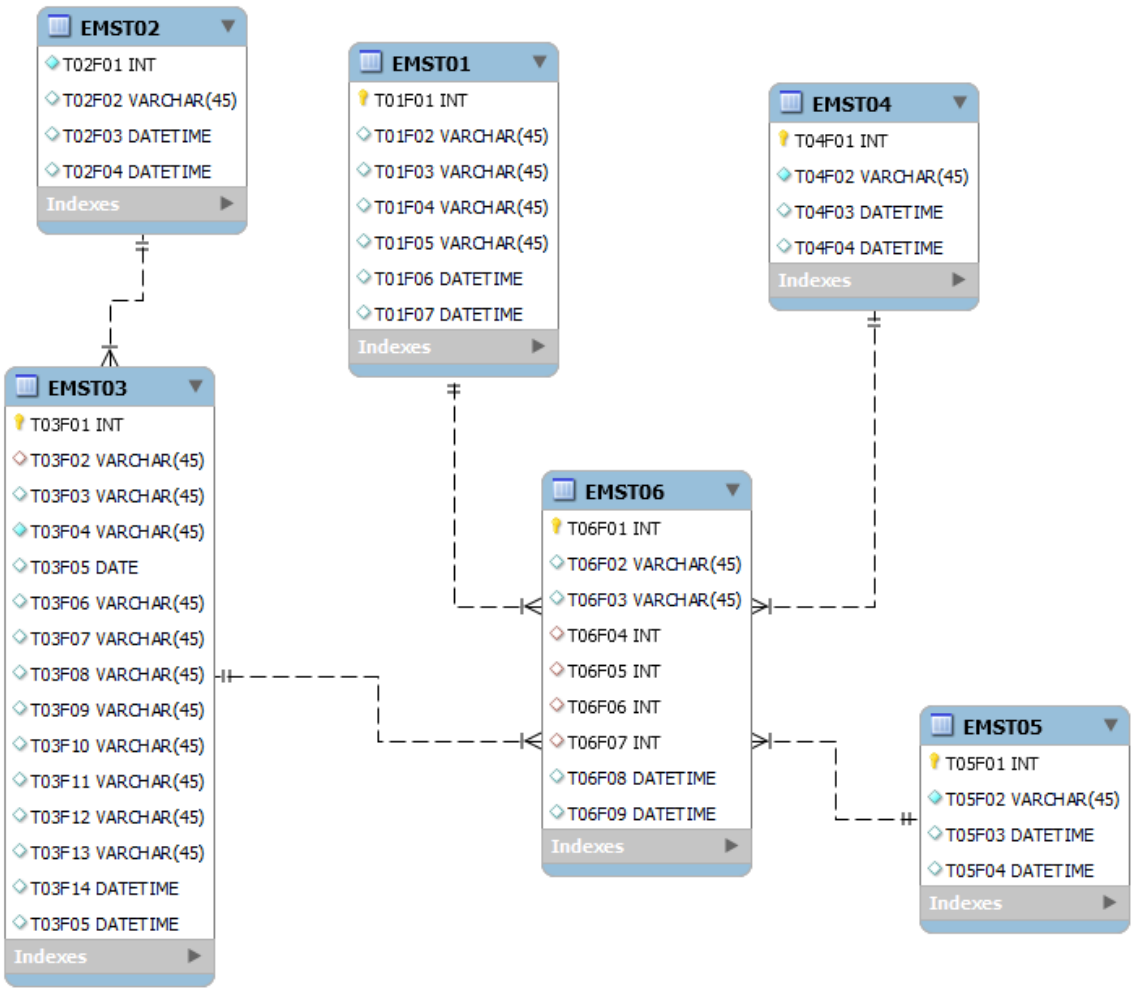

Fig. 5.2 (Database Design)

5.3 INPUT / OUTPUT AND INTERFACE DESIGN

5.3.1 Admin Module

The Admin module is a crucial part of the Employee Management System, as it provides the necessary functionalities for managing employee details, task assignments, and monitoring overall system operations. The Admin module typically includes the following features:

User Authentication and Authorization: The Admin module includes functionality for user authentication and authorization. This allows the system to ensure that only authorized administrators can access the system and perform administrative tasks.

System Configuration: The Admin module may include features for configuring system settings, such as managing user roles and permissions, setting up email notifications, and customizing system preferences.

User Management: The Admin module allows the administrator to manage user accounts, including adding, editing, and deleting user accounts. It may also include features for resetting passwords, managing user roles, and permissions.

System Maintenance: The Admin module may include features for system maintenance, such as backing up and restoring the database, managing system updates, and handling system errors or exceptions.

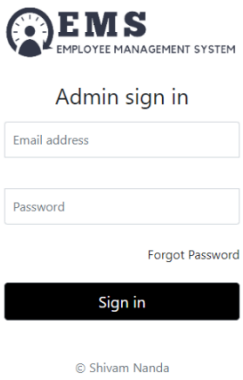

Fig. 5.3 (Admin Login)

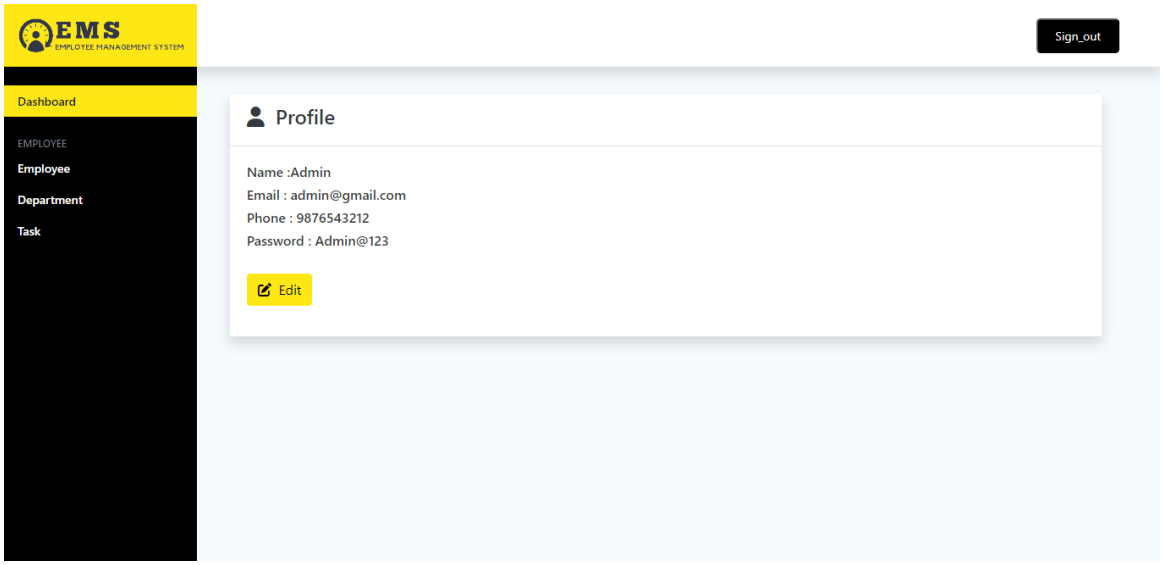

Fig. 5.4 (Admin Profile Module)

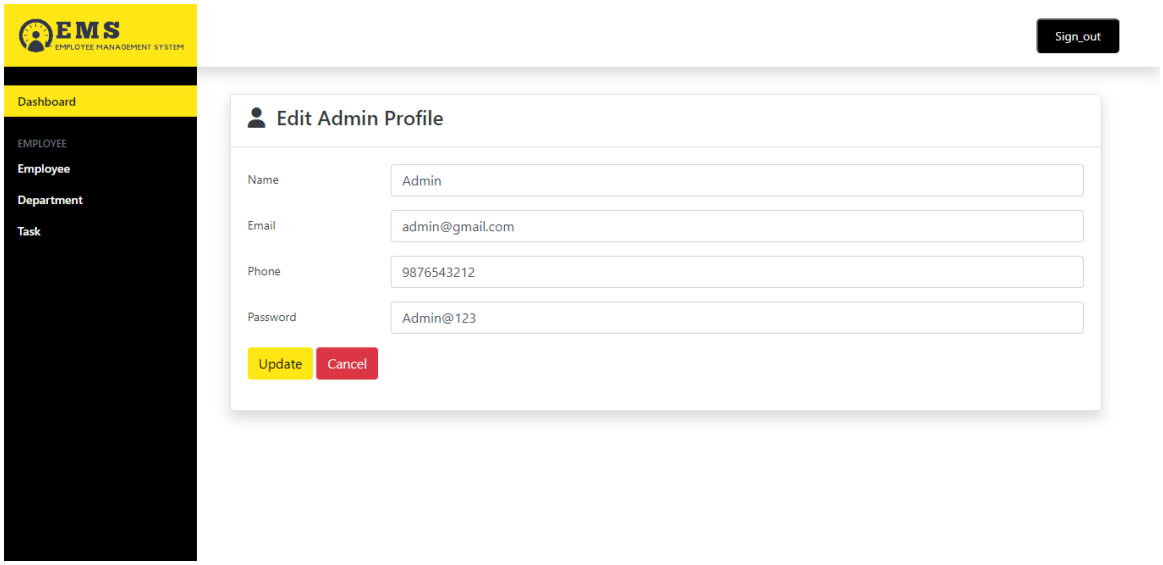

Fig. 5.5 (Admin Profile Edit Module)

5.3.2 Employee Management: The Admin module allows the administrator to manage employee details, including adding, editing, and deleting employee records. It also includes features for updating employee information, such as personal details, job titles, and department assignments.

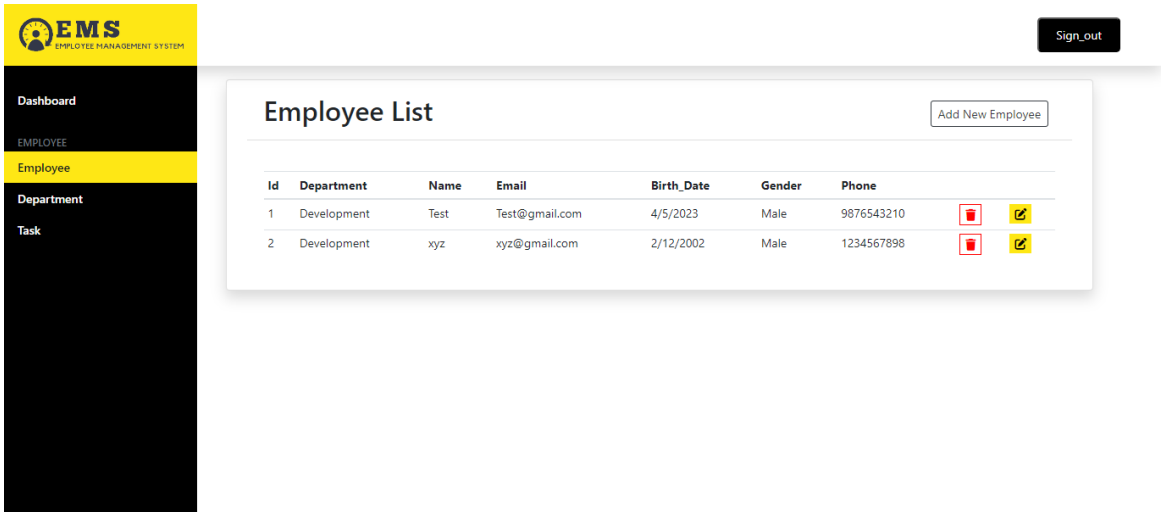

Fig 5.6 (Employee List Module)

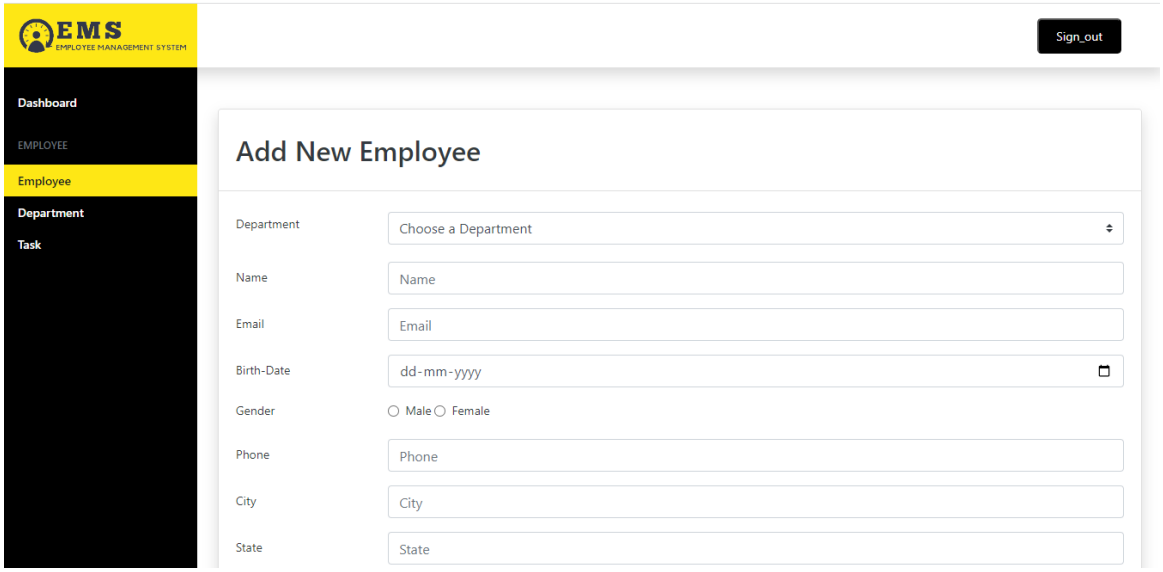

Fig. 5.7 (Add New Employee Module)

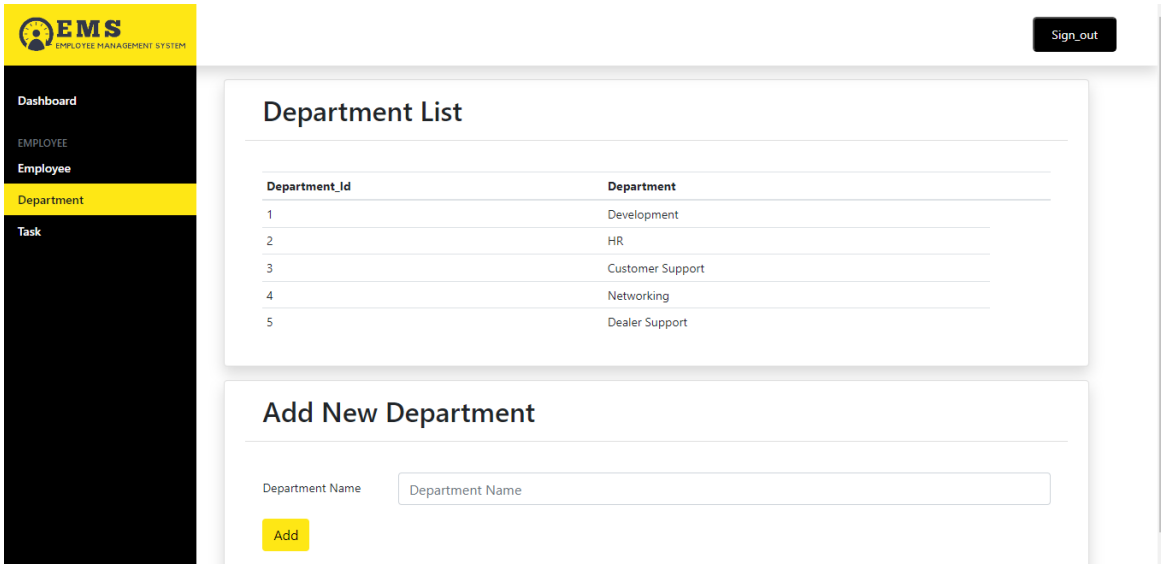

Fig. 5.8 (Department List and Manage Module)

5.3.3 Task Assignment: The Admin module allows the administrator to assign tasks to specific employees. This includes features such as creating new tasks, assigning tasks to employees, setting task deadlines, and tracking task progress.

Task Monitoring and Reporting: The Admin module provides functionality for monitoring and tracking task progress. This includes features such as viewing the status of assigned tasks, generating reports on task completion, and identifying overdue tasks.

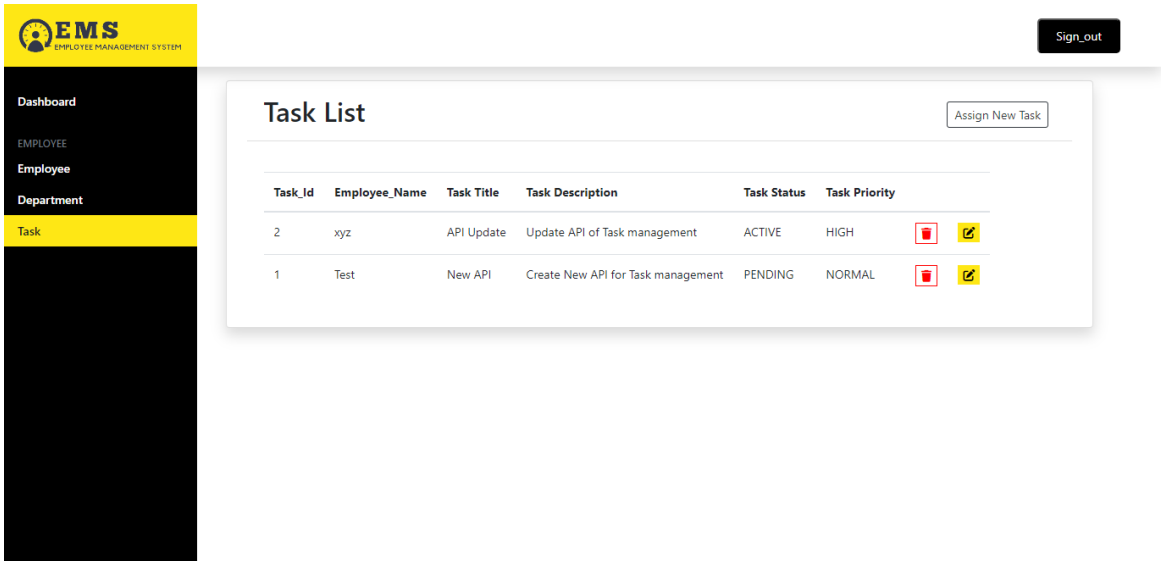

Fig. 5.9 (Task List Module)

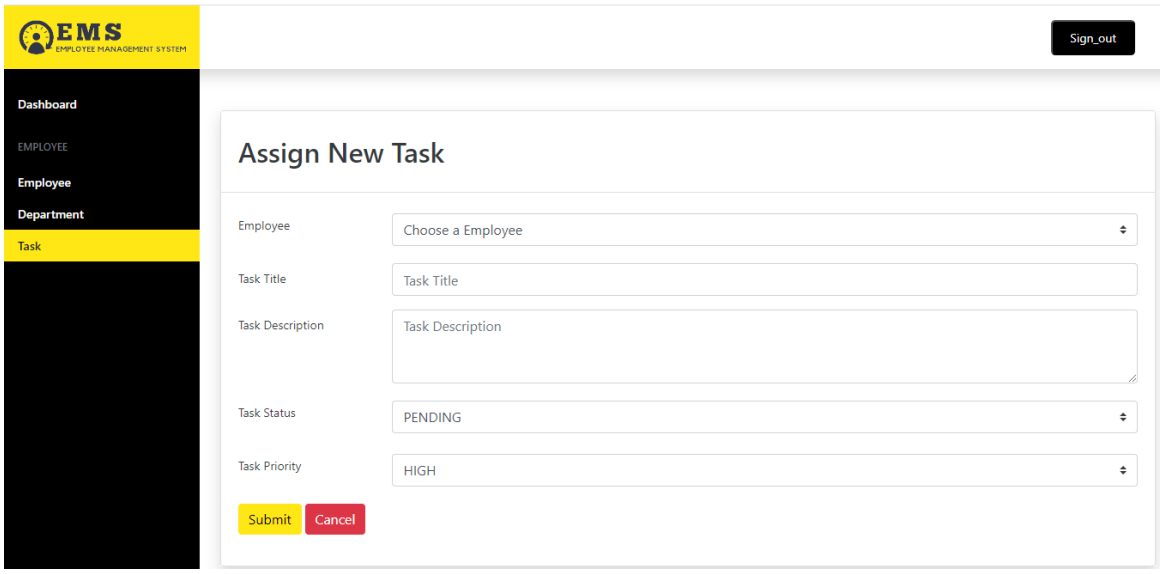

Fig. 5.10 (Assign New Task to Employee Module)

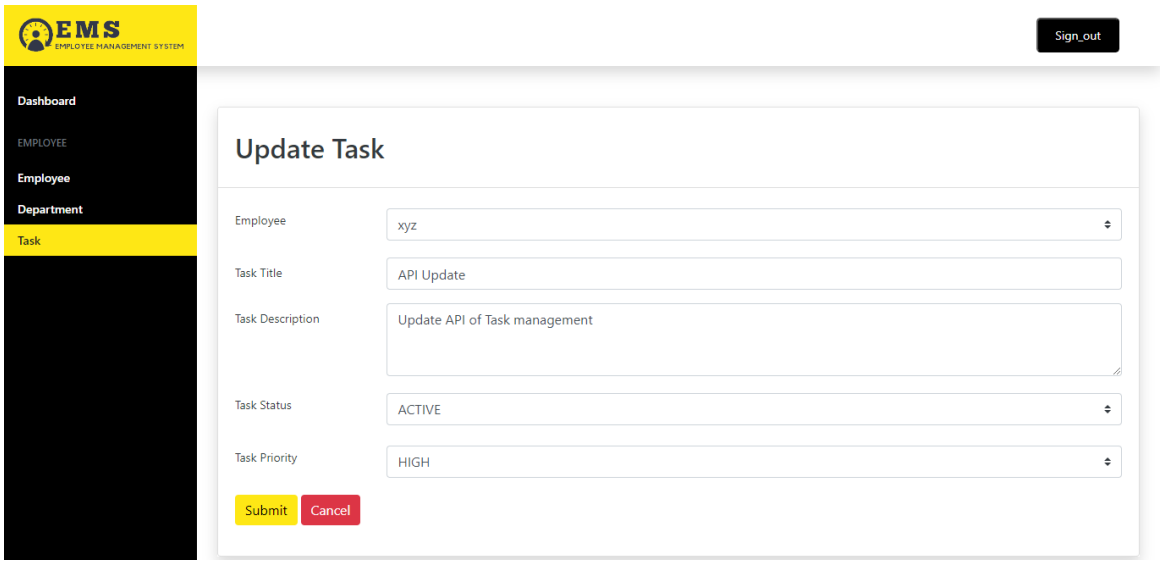

Fig. 5.11 (Update Task Status and Information)

CHAPTER 6 – IMPLEMENTATION

6.1 IMPLEMENTATION PLATFORM / ENVIRONMENT

Software Development Technology: .NET Framework & JavaScript

Development Tool: Visual Studio, Visual Studio Code, MySQL Workbench

6.2 MODULES SPECIFICATIONS

The Employee Management module is a critical component of the Employee Management System, which focuses on managing employee details and maintaining accurate records. Some of the specifications of the Employee Management module may include:

Employee Information Management: This module allows for the management of employee details, such as personal information, contact details, job title, department, and other relevant information. It provides features for adding, editing, and deleting employee records, as well as updating employee information as needed.

Employee Search and Filtering: The module may include features that allow for easy searching and filtering of employee records based on various criteria, such as name, job title, department, or other custom filters. This helps in quickly locating specific employee records and retrieving relevant information.

On the other hand, the Task Management module focuses on managing tasks assigned to employees and monitoring their progress. Some of the specifications of the Task Management module may include:

Task Assignment and Tracking: This module allows for the assignment of tasks to specific employees, setting task deadlines, and tracking task progress. It may include features such as creating new tasks, assigning tasks to employees, updating task status, and sending notifications for task deadlines.

Task Prioritization and Scheduling: The module may include features for prioritizing tasks based on their importance and urgency, as well as scheduling tasks based on resource availability and workload. This helps in efficiently managing tasks and ensuring timely completion.

The Employee Management and Task Management modules work together to ensure efficient management of employee details and task assignments, enabling smooth functioning of the Employee and Task Management System and improving overall productivity and performance.

CHAPTER 7 – TESTING

7.1 TESTING PLAN

A testing plan is a crucial component of software development that outlines the approach, scope, and objectives of the testing process. Here are some key elements that should be included in a testing plan for any software after development:

Test Objectives: Clearly define the objectives of testing, such as identifying defects, validating functionality, verifying compliance with requirements, and assessing performance and security.

Test Scope: Define the scope of testing, including the components of the software that will be tested, the environments in which testing will be conducted, and the testing techniques that will be used.

Test Schedule: Develop a timeline for testing activities, including milestones, deadlines, and dependencies, to ensure that testing is conducted in a timely and organized manner.

Test Resources: Identify the resources required for testing, such as personnel, hardware, software, and tools, and ensure that they are allocated appropriately for efficient testing. Test Techniques: Select appropriate testing techniques based on the nature of the software, such as functional testing, usability testing, performance testing, security testing, and compatibility testing.

Test Data: Define the test data that will be used for testing, including valid and invalid data, edge cases, and boundary values, to ensure comprehensive coverage of the software's functionality.

Test Environment: Set up a testing environment that mirrors the production environment as closely as possible, including the hardware, software, and network configurations, to ensure accurate testing results.

Test Scripts and Test Cases: Develop test scripts and test cases that cover all the defined test objectives and scope, including test inputs, expected outputs, and test procedures, to ensure systematic and comprehensive testing.

Test Execution: Conduct testing activities according to the defined schedule, using the selected techniques, test data, and test cases, and document the results of each test for analysis and review.

Defect Tracking and Reporting: Establish a process for tracking and reporting defects identified during testing, including their severity, priority, and resolution status, to ensure timely and effective resolution of issues.

Retesting and Regression Testing: Conduct retesting and regression testing after defect resolution or changes to the software to ensure that fixes or changes do not introduce new defects or impact existing functionality.

Test Completion and Documentation: Complete testing activities according to the defined schedule and criteria, and prepare comprehensive documentation of the testing process, including test results, test coverage, and any issues identified and resolved.

By following a well-defined testing plan, software can be thoroughly tested for quality, functionality, performance, and security, resulting in a more reliable and robust product that meets the needs of end users.

7.2 TEST METHOD

Unit Testing: This type of testing focuses on testing individual units or components of the software in isolation, typically at the code level. It helps identify defects or bugs in the smallest possible units of the software, such as functions or methods, and ensures that they are working as expected.

Integration Testing: Integration testing involves testing the interaction and integration between different components or modules of the software to verify that they work correctly when combined. It helps identify issues related to communication, data flow, and interface compatibility between different parts of the software.

System Testing: System testing is performed on the entire software system as a whole, including all integrated components and modules, to verify that it meets the specified requirements and functions as expected. It helps identify issues related to system behaviour, performance, and overall functionality.

Regression Testing: Regression testing is performed to ensure that changes or modifications in the software do not introduce new defects or impact the existing functionality. It involves retesting of previously tested functionalities to ensure their continued correctness after changes.

Performance Testing: Performance testing is conducted to assess the performance and scalability of the software under different loads and stress conditions. It helps identify issues related to response time, throughput, resource utilization, and overall system performance.

Security Testing: Security testing is performed to identify vulnerabilities and weaknesses in the software that may expose it to security risks. It includes testing for authentication, authorization, data encryption, and other security-related aspects to ensure the software's security.

User Acceptance Testing (UAT): UAT involves testing the software from the end user's perspective to ensure that it meets their requirements and expectations. It helps identify issues related to usability, user interface, and overall user experience.

7.2.1 TEST CASES (TEST ID, TEST CONDITION, EXPECTED OUTPUT, ACTUAL OUTPUT, REMARK)

Test ID	Test Condition	Expected Output	Actual Output	Remark
TC001	Create New Employee	New employee record is added successfully	New employee record is added successfully	Pass
TC002	Update Employee Details	Employee details are updated successfully	Employee details are updated successfully	Pass
TC003	Delete Employee	Employee record is deleted successfully	Employee record is deleted successfully	Pass
TC004	Search Employee	Employee details are displayed correctly	Employee details are displayed correctly	Pass
TC005	Assign Task to Employee	Task is assigned to the selected employee	Task is assigned to the selected employee	Pass
TC006	Update Task Status	Task status is updated successfully	Task status is updated successfully	Pass

TC007	**View Task List**	**Task list is displayed with correct details**	**Task list is displayed with correct details**	**Pass**
TC008	Invalid Login Credentials	Error message is displayed for invalid credentials	Error message is displayed for invalid credentials	Pass
TC009	Password Reset	Password is reset successfully	Password is reset successfully	Pass
TC010	Task Completion	Task is marked as completed with correct status	Task is marked as completed with correct status	Pass

CHAPTER 8 – CONCLUSION AND DISCUSSION

8.1 OVERALL ANALYSIS OF INTERNSHIP

The overall analysis of my internship as an intern in a company, where I learned both front-end and back-end technologies, can be described as a rewarding and enriching experience. During my internship, I gained practical hands-on experience in web development using front-end technologies such as HTML, CSS, and JavaScript for creating visually appealing and interactive user interfaces. I also learned back-end technologies such as databases, server-side scripting, and APIs for handling data management and server-side operations. Through various projects and assignments, I gained a deep understanding of the entire software development life cycle and learned how to work with modern web development frameworks and tools. The internship also provided me with opportunities to collaborate with a team, solve real-world problems, and apply best practices in coding and software development. Overall, the internship equipped me with valuable technical skills, enhanced my understanding of front-end and back-end technologies, and prepared me for a successful career in the field of web development.

8.2 DATES OF EVALUATION

Evolution Date: 20th April 2023

8.3 SUMMARY OF INTERNSHIP WORK

In January 2023, I began my internship at RKIT, working out of the Rajkot ON office. I found inspiration in my everyday journeys, thanks to the fantastic work environment and the pleasant, helpful, customer-service focused, compassionate, and truthful personnel. It was an opportunity for me to demonstrate my worth as an employee, a trustworthy co-worker, and a dedicated student. It also provided me with crucial office experience that I had lacked in my previous jobs.

One of the most significant abilities I learned during my internship was versatility. Working across multiple technologies is typical in corporations, and that is exactly what I did while working on an Online Bidding Platform project. As I completed more tasks, I was given additional responsibilities, and I was always excited to receive new assignments. The encouragement and honest feedback I received from my Protium co-workers were more than enough to put me at ease. I am truly grateful to all of my co-workers at RKIT for giving me the opportunity to grow both personally and professionally.

8.4 LIMITATION AND FUTURE ENHANCEMENT

Limitations of Employee & Task Management System:

- Lack of real-time updates
- Limited scalability
- Dependency on technology
- User interface/usability issues
- Lack of integration with other systems

Future Enhancements of Employee & Task Management System:

- Real-time notifications and updates
- Enhanced scalability and performance
- Adoption of modern technologies
- Improved user interface/usability
- Integration with other systems
- Advanced reporting and analytic

REFERENCES

1. https://learn.microsoft.com/en-us/dotnet/framework/
2. https://developer.mozilla.org/en-US/docs/Web/JavaScript
3. https://api.jquery.com/
4. https://getbootstrap.com/docs/4.1/getting-started/introduction/
5. https://www.tutorialsteacher.com/webapi
6. https://www.w3schools.com/mysql/