WIREPOOL

A PROJECT REPORT

Submitted by,

Mr. Kunal Rana - 20201CSE0002
Mr. Yash Choudhary -20201CSE0010
Ms. Manshi Maurya- 20201CSE0029
Mr. Aryan Kaushik- 20201CSE0057
Ms. Eshana Chauhan- 20211LCS0001

Under the guidance of,

Dr. Pamela Vinitha Eric

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING, COMPUTER ENGINEERING, INFORMATION SCIENCE AND ENGINEERING Etc.

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING & INFORMATION SCIENCE

CERTIFICATE

This is to certify that the Project report **"WIREPOOL"** being submitted by "Kunal Rana, Yash Choudhary, Manshi Maurya, Aryan Kaushik, Eshana Chauhan" bearing roll number(s) "20201CSE0002, 20201CSE0010, 20201CSE0029, 20201CSE0057, 20211LCS0001" in partial fulfillment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

Dr. Pamela Vinitha Eric
Prof-CSE
School of CSE & IS
Presidency University

Dr. Pallavi R
Assoc. Prof & HoD
School of CSE & IS
Presidency University

Dr. C. KALAIARASAN
Associate Dean
School of CSE & IS
Presidency University

Dr. SHAKKEERA L
Associate Dean
School of CSE & IS
Presidency University

Dr. SAMEERUDDIN KHAN
Dean
School of CSE & IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING & INFORMATION SCIENCE

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **WIREPOOL** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr. Pamela Vinitha Eric, School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Kunal Rana- 20201CSE0002
Yash Choudhary- 20201CSE0010
Manshi Maurya- 20201CSE0029
Aryan Kaushik- 20201CSE0057
Eshana Chauhan- 20211LCS0001

ABSTRACT

This project presents a novel mobile application developed on the MERN stack, serving as an integrated service platform that bridges end users with a wide array of professionals, including OEMs, Process Specialists, Electricians, and Freelance Consultants. The mobile app offers an intuitive and user-friendly interface, facilitating on-demand access to expert services. Users can seamlessly connect with skilled professionals across diverse domains, enhancing efficiency and convenience in obtaining specialized assistance. The objective is to streamline the process of accessing expert services, providing a comprehensive solution to meet user needs. By creating a unified platform, this project aims to elevate user experiences, offering a seamless and efficient way for individuals to tap into a network of professionals, ultimately redefining how expert services are accessed and availed through a modern and accessible mobile application.

ACKNOWLEDGEMENT

LIST OF FIGURES

TABLE OF CONTENTS

List of Abbreviations

DOM   Document Object Model

W3C   World Wide Web Consortium.

ISO   International Organization for Standardization

NPM   Node Package Manager

API   Application Program Interface

HTML   Hypertext Markup Language

HTTP   Hypertext Transfer Protocol

URL   Uniform Resource Locator

JSON   JavaScript Object Notation

JS   JavaScript

JSX   JavaScript XML

XML   Extensive Markup Language

CHAPTER-1

INTRODUCTION

OVERVIEW

This report presents the comprehensive development and deployment of Wirepool, an innovative service-based application tailored to address the challenges encountered in service evaluation and management. Wirepool is designed to streamline service duration estimation and pricing mechanisms, catering to the dynamic needs of users.

Certainly, technology has undeniably become a crucial tool in contemporary online marketing. Nevertheless, a notable trend in Vietnam reveals the persistence of numerous small shops and grocery stores clinging to traditional offline business models. This approach may lead to suboptimal experiences for both buyers and sellers. For instance, sellers may have desirable products, but potential buyers remain unaware of them. Additionally, buyers might urgently need a particular item, only to find the local store out of stock.

Contrastingly, online shopping presents a myriad of advantages. It allows customers to explore a wide array of products, compare prices effortlessly, and make informed decisions. Embracing technology in commerce not only facilitates smoother transactions but also enhances the overall shopping experience for both consumers and sellers. It's crucial for businesses to adapt to the evolving landscape to better cater to the dynamic needs of the market.

In order to make a mobile application that can acquire the needs of both customers and retailers, MERN (MongoDB, Express.js framework, ReactJS library, NodeJS platform) is one of the powerful stacks that can help us to develop an e-commerce web application.

PROBLEM STATEMENT

In the dynamic landscape of industry and services, the need for efficient collaboration between end users and skilled professionals has never been more crucial. "Wirepool" emerges as a comprehensive solution, providing a seamless digital platform to connect end users with Original Equipment Manufacturers (OEMs), Process Specialists, Electricians, and Freelance Consultants. This application aims to streamline service requests, foster real-time communication, and enhance the overall user experience. With an intuitive interface, robust authentication, and a diverse array of services, We envision a future where expertise and demand harmonize, creating a thriving ecosystem of seamless service delivery. Join us on this journey to revolutionize how services are accessed, delivered, and experienced.

E-COMMERCE

Definition:

EC for short (E-commerce) is a concept referring to transactions, purchases, and sales of goods and services via the Internet.

E-commerce was first known in the 1960s. After years of development, as mobile devices became popular, social media increasingly affirmed the power and the boom of the webpage. Launchers promote the rapid development of commerce (E-commerce).[1]

Types:

Presently, various forms of e-commerce exist, encompassing fundamental categories such as

1. B2B (Business to Business): This involves trade transactions between companies, businesses, and organizations. Approximately 80% of contemporary e-commerce operates within this sphere.

2. B2C (Business to Consumer): This pertains to internet-based businesses directly providing goods and services to end consumers.

3. C2B (Consumer to Business): In this model, consumers sell their products or services to businesses or organizations.

4. C2C (Consumer to Consumer): This scenario unfolds when a consumer sells their goods or services directly to another consumer.

While there are additional models like G2C, G2B, etc., they are less frequently employed compared to these fundamental four forms.

Advantages:

1. Global Market Access: When establishing a physical store, your reach is inherently limited to a small geographic area. E-commerce effectively addresses this constraint by facilitating rapid market expansion. This broader reach, compared to traditional

direct sales, allows for easy introduction, purchase, and sale of products and services through both retailers and online markets.

2. Constant Accessibility: E-commerce operates 24/7, 365 days a year, providing a perpetual online presence. This constant accessibility creates significant opportunities for businesses to enhance sales continuously.

3. Cost Efficiency: E-commerce significantly reduces operational costs in comparison to traditional commerce. Expenses such as booth rentals, sales personnel, and management are notably more economical. Savings in operational costs empower sellers to provide customers with more incentives and better discounts, fostering a mutually beneficial relationship.

4. Inventory Optimization: Electronic tools streamline ordering, delivery, and payment processes, leading to substantial cost savings and reduced inventory for e-commerce businesses.

5. Precision in Customer Marketing: E-commerce entities leverage customer data and behavior tracking to swiftly identify and market products and services tailored to individual consumer preferences. This personalized approach enhances customer satisfaction.

6. Flexible Work and Purchase Locations: Operating an e-commerce business liberates individuals from the confines of a physical office, while buyers can make purchases without the need to visit a physical store. All that's required for both parties is an internet-connected device. This mutual convenience epitomizes the flexibility offered by e-commerce.[2]

Challenges:

1. Internet Access Requirement: Participation in e-commerce necessitates a device connected to the Internet for buying and selling. While a significant portion of the population now has internet access, its availability remains limited in many areas.

2. Trust Concerns: Products and services lacking a tangible presence pose challenges for cautious buyers who prefer trying before purchasing. Doubt among both buyers and sellers often results in incomplete transactions, particularly if they have encountered untrustworthy partners in the past.

3. Payment Method Limitations: In the current online retail landscape in Vietnam, the predominant payment method involves receiving goods before payment. Although the payment gateway system in Vietnam is advancing, it still lacks the reliability necessary for users to adopt it as their primary payment method, contributing to ongoing challenges.

Moreover, the e-commerce sector encounters various other challenges, encompassing technical issues, competition, and payment-related concerns, among others.

CHAPTER – 2

LITERATURE SURVEY

Authors	Year	Research Paper Title	Journal / Article Name Paper Published in	Objective	Key Findings
[3]Dr. Santosh Kumar Shukla, Shivam Dubey, Tarun Rastogi, Nikita Srivastava	2022	Application using MERN Stack	International Journal for Modern Trends in Science and Technology	eCommerce is transforming B2B enterprises by expanding reach, cutting costs, and enhancing customer experiences. The goal for businesses is clear: define digital strategies to connect with new clients, especially those historically challenging or costly to engage, thus fostering growth beyond existing models.	The ubiquity of e-commerce in today's digitally connected world underscores its significance for buying and selling products and services, with digital carts becoming an integral part of daily life. This project, developed on the MERN stack, seamlessly integrates digital payment gateways, product sorting, and search functionalities, while utilizing MongoDB for efficient data storage and retrieval. In the current landscape, B2B and B2C commerce thrive, and our proposed model outperforms previous ones, delivering superior results for both

					consumers and retailers. As e-commerce continues to evolve, the need for seamless, user-friendly platforms becomes paramount, and our project addresses these requirements effectively.
[4]Vipul Kaushik, Kamali Gupta, Deepali Gupta	2018	React Native Application Development	Social Science Research Network		Creating mobile apps compatible on cross platforms is challenging, requiring developers to be familiar with multiple native platforms. Existing hybrid mobile application frameworks often fail to provide a consistent user experience across different platforms. The research work presented in this paper utilizes the advantages of the React-Native framework to create a hybrid mobile application called "Medikyte app" that focuses on medication solutions.

					The framework has been developed for both Android and iOS platforms, and the results show a satisfactory user experience on both platforms. The proposed methodology for developing the Medikyte app utilizes JavaScript ES6
[5]Qui Chong	2019	Service supply and demand matching method and system	Patent	The objective of the service supply-demand matching method and system is to provide a platform for demanders and suppliers to connect in real-time, facilitating the matching of demand services with supply services	The service supply-demand matching method and system aim to establish a platform for real-time connection between demanders and suppliers, facilitating the matching of demand services with supply services. The method involves extracting and matching keywords of the demand service, successfully sending the demand application to the matched supplier, and providing remuneration to the supplier. The system

					focuses on establishing a cooperation relationship between the demander and the supplier by sharing contact information and collecting remuneration from the demander upon completion of the cooperation. The overall objective is to enable the demand side to find solutions from the supply side, allowing the supply side to have more partners and employment opportunities.
[6]Buddhini Gayathri Jayatilleke, Gaya R. Ranawaka, Chamali Wijesekera and Malinda C.B. Kumarasinha	2018	Development of mobile application through design-based research	Asian Association of Open Universities Journal	The research objectives include addressing the scarcity of mobile learning research, designing a mobile solution for OUSL's Faculty of Health Sciences, testing it with various stakeholders, documenting findings, and formulating guiding principles for mobile learning.	The study found that stakeholder-centered design-based research effectively developed a mobile learning app through concurrent evaluation and testing phases. Challenges in development included time, cost, technical optimizations, and teaching limitations. Adequate support

				The study aims to contribute insights into innovative teaching practices, emphasizing the importance of design-based research in educational technology development	structures are vital for sustaining innovation in mobile learning. The app will undergo further evaluation with actual students, and the study offers guiding principles for mobile solution design and research. Consideration of target audience, content, and organizational culture is essential, emphasizing cultural dimensions.
[7]Thomas C. G. & A. Jayanthila Devi	2021	A Study and Overview of the Mobile App Development Industry	International Journal of Applied Engineering and Management Letters (IJAEML)	To analyze and understand the Mobile App Development Industry worldwide and in India, to understand the preference, usage, revenue, cost, and scope of App Development in India and to perform a SWOC Analysis for the Mobile App Development Industry.	The study emphasizes the crucial phase of setting clear objectives for app development, ensuring a comprehensive understanding of technical and non-technical requirements. Key questions regarding the target audience, app purpose, benefits, technologies, competitors, and Unique Selling Point (USP) must be addressed. Long-term vision

					is deemed essential for the app's success in an ever-evolving technological landscape. Following objective setting, the focus shifts to app design and prototyping, emphasizing wireframing for effective UI design, Navigation Design, Interface Design and functionality. The backend selection is a critical decision, offering choices between custom servers, cloud servers, and Mobile Backend as a Service (MBaaS) solutions. Each option comes with distinct challenges and benefits, underscoring the importance of thoughtful decision-making in the app development process.
[8]Laith T. Khrais 1 Abdullah M. Alghamdi	2021	The Role of Mobile Application Acceptance in Shaping E-Customer Service	Journal of Retailing and Consumer Services	This research aims to precisely assess the impact of mobile app acceptance on e-commerce	This research underscores the pivotal role of mobile app acceptance in enhancing the customer experience

					service delivery, investigate factors propelling the adoption of mobile apps in online retail, and scrutinize how mobile app acceptance specifically shapes e-customer service. The study also seeks to address targeted research questions regarding the influence of mobile app acceptance on e-customer service, customer experience, and engagement. By pinpointing challenges and effective strategies related to mobile app acceptance in e-commerce, the research endeavors to offer streamlined insights for retailers to strategically leverage technology, ensuring heightened customer	within the e-commerce sector. The study reveals that the positive impact of mobile apps on service delivery, customer engagement, and overall satisfaction is driven by factors such as convenience, ease of use, and swift transactional processes. Interviews and questionnaires demonstrate that mobile app adoption is reshaping e-customer service dynamics, leading to improved interactions, faster issue resolution, and greater customer empowerment through self-service features. While emphasizing the transformative potential of mobile apps in meeting evolving customer expectations, the research suggests a need for future exploration of

				satisfaction and sustained competitiveness in the dynamic digital marketplace.	challenges and strategies associated with mobile app acceptance in e-commerce. Overall, the study advocates for the integration of innovative technologies to stay competitive in the rapidly evolving e-commerce landscape.
[9]P.K. Paul & Sreeramana Aithal	2019	MOBILE APPLICATIONS SECURITY : AN OVERVIEW AND CURRENT TREND	IEEE Communications magazine	The paper aims to explore and emphasize the importance of Mobile Security in Information Technology, focusing on safeguarding mobile services and devices from potential vulnerabilities and attacks, contributing insights to the field of Information Assurance.	The key finding of the paper is the growing significance of Mobile Security in Information Technology, highlighting the need to protect mobile services and devices from potential vulnerabilities and attacks, particularly in the context of wireless security and mobile application concerns. The paper emphasizes the increasing relevance of mobile security within the broader landscape of Information Assurance.

[10]Dan Johansson, Karl Andersson	2015	Mobile e-Services: State of the Art, Focus Areas, and Future Directions	International Journal of E-Services and Mobile Applications,	The objective of the research paper is to explore and analyze e-services from a mobility perspective. This paper aims to: Position mobile e-services within the research field. Review related work on mobile e-services. Present and examine existing challenges and opportunities when combining mobility and e-services. Discuss important focus areas and future directions concerning mobile e-services, emphasizing challenges and opportunities in areas like acceptance and adoption, availability anytime and anywhere, and cooperation.	Mobile e-services are considered the next generation of internet-based services, surpassing fixed network computing in many aspects. Statistics reveal a significant increase in the number of active mobile broadband subscriptions compared to fixed subscriptions, indicating the dominance of mobile computing. The paper highlights the four types of mobility: terminal mobility and three additional types, suggesting the extensive nature of mobility in computing. It emphasizes the importance of understanding and addressing challenges related to mobile e-services, including acceptance, adoption, availability, and cooperation, to

					enhance e-participation.
[11]Md. Abdur Razzaque, Amitava Chatterjee	2014	Mobile computing and its applications in healthcare	Journal of Informatics and Communication Technology (JICT)	Investigate the integration of mobile computing technologies in the healthcare sector. Explore the evolution, significance, and implications of mHealth applications. Evaluate the efficacy of mobile applications designed for healthcare purposes. Assess the broader impact of mobile computing on healthcare services and accessibility.	Enhanced accessibility to healthcare services through mobile platforms. Advancements in mHealth technologies facilitating remote patient care. Improved patient care through real-time communication and monitoring. Identification of challenges (e.g., data security) and opportunities for healthcare transformation through standardized regulations and technological advancements.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

Compatibility Issues:
- Lack of uniformity across different platforms leads to challenges in compatibility and consistent user experience.
- Existing hybrid mobile application frameworks may fail to provide a seamless user experience across various platforms.

Technical Limitations:
- Development complexities arise when targeting multiple native platforms simultaneously.
- Technical optimizations may pose challenges, leading to time and cost constraints in application development.

Acceptance and Adoption:
- Resistance to the adoption of new mobile services among users and stakeholders.
- Challenges related to the acceptance and adoption of mobile apps in specific industries, hindering their full potential utilization.

Regulatory and Compliance Challenges:
- Adherence to industry-specific regulations and compliance standards in healthcare applications might present challenges during development and deployment.
- Ensuring alignment with privacy laws and regulatory frameworks for healthcare data handling and storage.

Technical Limitations in Mobile App Security:
- Continuous evolution of threats and vulnerabilities necessitates ongoing updates and security patches, posing challenges for maintaining robust security measures.
- Limitations in effectively addressing sophisticated cybersecurity threats targeting mobile applications.

CHAPTER-4

PROPOSED METHODOLOGY

The methodology for implementing "Wirepool" involves the following key components:

1. User-Centric Design:

Develop an intuitive and user-friendly interface for the mobile app to ensure accessibility and ease of use.
Prioritize user experience by incorporating feedback and insights from end users throughout the design process.

2. Diverse Service Integration:

Collaborate with Original Equipment Manufacturers (OEMs), Process Specialists, Electricians, and Freelance Consultants to onboard a diverse pool of skilled professionals.
Implement a systematic process for verifying and validating the credentials and expertise of the service providers.

3. Real-Time Communication Features:

Integrate real-time communication features within the app to facilitate seamless interactions between end users and service providers.
Implement instant messaging, video calls, and other communication tools to enhance collaboration and problem-solving.

4. Streamlined Service Requests:

Develop a streamlined system for end users to submit service requests, specifying their requirements and preferences.
Implement algorithms to efficiently match service requests with the most suitable professionals based on expertise, location, and availability.

5. Robust Authentication and Security:

Prioritize the security of user data and transactions by implementing robust authentication measures.
Utilize encryption protocols to safeguard sensitive information and ensure a secure environment for both end users and service providers.

6. Continuous Improvement:

Establish feedback loops and analytics tools to continuously monitor and analyze user interactions and service delivery.
Regularly update the app based on user feedback and technological advancements to enhance functionality and address any issues that may arise.

7. Promotion and Community Building:

Implement a strategic marketing plan to promote the Wirepool platform to both end users and service providers.
Foster a sense of community within the platform to encourage collaboration, knowledge-sharing, and positive interactions among users.

8. Scalability and Adaptability:

Design the platform with scalability in mind to accommodate the potential growth in user base and service offerings.
Ensure adaptability to industry changes and emerging technologies, allowing Wirepool to stay at the forefront of service delivery innovation.

CHAPTER-5

OBJECTIVES

- **Enhance Service Accessibility:**

Provide a centralized hub where end users can easily discover, explore, and access a wide range of services tailored to their specific needs.

- **Promote Real-Time Communication:**

Foster effective communication channels within the application, enabling seamless interactions between end users and service providers for better collaboration and understanding.

- **Integrate Payment Solutions:**

Facilitate secure and convenient transactions by integrating reliable payment gateways, ensuring a smooth and trustworthy payment process for both end users and service providers.

- **Ensure Scalability and Performance:**

Develop a scalable architecture that can accommodate the growth of users and services over time while maintaining high performance and responsiveness.

CHAPTER -6

REQUIREMENT ANALYSIS

1. Hardware Components

2. Software Components

1. HARDWARE COMPONENTS

1. Server Infrastructure:

Physical Servers or Cloud Infrastructure: Depending on the scale of the application, you might opt for physical servers housed in a data center or utilize cloud infrastructure services such as Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure. Cloud platforms offer scalability, flexibility, and robust security features.

Server Hardware Specifications:

Processor: Multi-core processors (e.g., Intel Xeon or AMD EPYC) to handle concurrent requests and data processing.

RAM: Adequate RAM (e.g., 64GB or more) to ensure smooth operations and accommodate in-memory data caching.

Storage: SSD storage for faster data retrieval and lower latency. Consider RAID configurations for data redundancy and fault tolerance.

2. Database Servers:

Database Management System (DBMS): MongoDB, being the chosen database for the MERN stack, requires dedicated servers or instances for storing, retrieving, and managing data.

Server Hardware Specifications:

Processor: Efficient processors to handle complex queries and data operations.

RAM: Sufficient memory to support database operations, caching, and indexing.

Storage: High-speed SSD storage with ample capacity to store user data, service listings, transaction details, etc.

3. Networking Components:

Load Balancers: Distribute incoming application traffic across multiple servers to ensure high availability and reliability.

Firewalls: Hardware firewalls to monitor and control incoming and outgoing traffic, enhancing the application's security posture.

Routers and Switches: Manage network traffic, facilitate communication between servers, and ensure seamless data transfer.

4. Storage Solutions:

Network-Attached Storage (NAS) or Storage Area Network (SAN): For centralized storage, backup, and data retrieval. These solutions offer scalability and high availability.

Backup Devices: Tape drives, external hard drives, or cloud-based backup solutions to create regular backups of critical application data.

5. Client Devices:

End-user Devices: Devices such as smartphones, tablets, or computers used by end-users to access the Wirepool application. Ensure compatibility and optimal performance across various devices and operating systems.

Peripheral Devices: Accessories like printers, scanners, or biometric devices, if required for specific functionalities within the application.

6. Security Components:

Security Appliances: Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), and other security appliances to monitor, detect, and mitigate potential security threats.

SSL Certificates: Secure Socket Layer (SSL) certificates to encrypt data transmitted between clients and servers, ensuring data privacy and integrity.

2. SOFTWARE COMPONENTS

1. Operating Systems:

Server OS: Choose a stable and secure server operating system like Linux distributions (e.g., Ubuntu Server, CentOS) or Windows Server, based on application compatibility and development preferences.

Client OS: Ensure the Wirepool application is compatible with popular client operating systems, including iOS, Android, Windows, and macOS.

2. Development Frameworks and Libraries:

Front-end Framework:

React Native: Enables the development of native mobile applications for both iOS and Android using JavaScript and React.

Redux: Helps manage application state in complex React Native applications.

Back-end Framework:

Node.js: A runtime environment that executes JavaScript on the server side.

Express.js: A fast, unopinionated, and minimalist web framework for Node.js, facilitating the creation of robust APIs.

Database Framework:

MongoDB: A NoSQL database used to store and manage application data. Integration with Mongoose, an Object Data Modeling (ODM) library for MongoDB, can simplify data operations.

3. Database Management and Tools:

MongoDB Atlas: A cloud-based database service that provides automated backups, scaling, and monitoring of MongoDB databases.

MongoDB Compass: A GUI tool to visualize and interact with MongoDB data, facilitating database management and query execution.

4. Middleware Components:

Authentication and Authorization:

Passport.js: Middleware for Node.js that provides authentication strategies.

JWT (JSON Web Tokens): A compact, URL-safe means of representing claims to be transferred between two parties, ensuring secure data transmission.

Caching:

Redis: An in-memory data structure store used as a cache or message broker, enhancing application performance by reducing database load.

5. API and Integration Components:

RESTful APIs: Design and implement RESTful APIs using Express.js to facilitate communication between the client-side application and server-side logic.

Third-Party Integrations: Integrate necessary third-party services or APIs for functionalities such as payment processing, notifications, or geolocation services.

6. Testing and Quality Assurance Tools:

Jest and Mocha: Testing frameworks for JavaScript, enabling the creation of unit tests and ensuring code reliability.

Postman: A popular tool for testing APIs, facilitating API development, and ensuring endpoints function as expected.

7. Security Components:

SSL/TLS Certificates: Implement SSL/TLS encryption to secure data in transit and establish

a trusted connection between clients and servers.

Helmet.js: A middleware for Express.js that helps secure Express applications by setting various HTTP headers.

OAuth 2.0: Implement OAuth 2.0 authentication for secure and authorized access to the application's resources.

8. Deployment and Continuous Integration/Continuous Deployment (CI/CD) Tools:

Docker: Containerization tool to package the application and its dependencies into a standardized unit, ensuring consistent environments from development to production.

Kubernetes: Orchestration platform for managing containerized applications, facilitating deployment, scaling, and management of application containers.

Jenkins or GitLab CI/CD: CI/CD tools to automate the software delivery pipeline, ensuring efficient code integration, testing, and deployment processes.

VISUAL STUDIO:

Visual Studio is an integrated development environment (IDE) created by Microsoft that provides a comprehensive set of tools and features for software development. It offers a rich and user-friendly environment for building a wide range of applications, including desktop applications, web applications, mobile apps, cloud-based solutions, and more.

Visual Studio supports multiple programming languages, including C#, Visual Basic, C++, F#, Python, JavaScript, and TypeScript. It provides developers with a unified interface and a suite of powerful tools to write, debug, test, and deploy their code efficiently.

Key Features of Visual Studio:

- Code Editor: Visual Studio offers a robust code editor with features like syntax highlighting, code completion (IntelliSense), code refactoring, and customizable keyboard shortcuts. It enhances productivity and helps developers write clean and error-free code.
- Debugging and Diagnostics: Visual Studio includes a powerful debugging toolset

that allows developers to step through code, set breakpoints, inspect variables, and analyze runtime behavior. It helps identify and fix bugs and issues during the development process.

- Integrated Testing Tools: Visual Studio provides built-in support for various testing frameworks and tools, enabling developers to write and execute unit tests, perform automated UI testing, and conduct performance profiling to optimize application performance.

- Integrated Version Control: Visual Studio integrates with popular version control systems like Git and Team Foundation Version Control (TFVC). It allows developers to easily manage source code, track changes, collaborate with team members, and handle branching and merging operations.

- Project and Solution Management: Visual Studio provides a flexible project and solution management system. Developers can create and manage multiple projects within a solution, organize files and dependencies, and control project settings and configurations.

- Extensibility and Marketplace: Visual Studio supports extensibility through a wide range of extensions and add-ons. Developers can enhance the IDE with additional tools, frameworks, and language support by accessing the Visual Studio Marketplace, which hosts a vast collection of community-created extensions.

- Cloud and Web Development: Visual Studio includes tools for developing cloud-native applications using Azure services, as well as web development tools for building modern web applications, including support for frameworks like ASP.NET and Node.js.

- Collaboration and Team Development: Visual Studio facilitates collaborative development by providing features such as Live Share, enabling real-time code sharing and collaboration among team members. It also integrates with project management tools like Azure DevOps for streamlined team coordination.

Visual Studio is widely used by developers worldwide due to its comprehensive set of features, flexibility, and extensive ecosystem. It caters to the needs of individual developers, small teams, and large enterprise-scale projects, making it a popular choice for software development across various platforms and industries.

CHAPTER – 7
MERN STACK

JavaScript

JavaScript is a scripting, object-oriented, cross-platform programming language that allows the connection of host environment objects and defines methods for their operation. It comes equipped with standard libraries containing objects like Array, Date, and Math, as well as fundamental programming language components such as managers, control frameworks, and statements.[12]

Extending JavaScript involves adding objects, enabling its application in various contexts, including:

- Client-side JavaScript: Employing objects to control browsers and the Document Object Model (DOM). Client-side extensions empower applications to manipulate HTML page components and respond to user actions like mouse hovers, form input, and page transitions.
- Server-side JavaScript: Introducing additional objects necessary for running JavaScript on the server. Server-side extensions allow applications to interact with databases, transfer data between requests, and execute functions in other parts of the application.

JavaScript, initially named ECMAScript in 1996, has undergone evolution with ECMAScript 2 (1998) and ECMAScript 3 (1999), ultimately evolving into the present JavaScript that functions seamlessly across browsers and devices. As an open standard language, organizations can utilize JavaScript for their applications. The ECMAScript Standard is a component of the ECMA-262 specification, approved by ISO as ISO-16262. Notably, the ECMAScript standard does not encompass descriptions for the Document Object Model (DOM), which is standardized by the W3C. The DOM defines how scripts display HTML objects, providing insight into the distinctive innovations employed in JavaScript programming.

NodeJS

Node.js is an open-source, system application, and server-side runtime environment. It provides an independent platform for the swift development of network applications, leveraging Chrome's JavaScript Runtime and utilizing the Google V8 JavaScript engine for code execution. A substantial portion of critical modules is scripted in JavaScript.[13]

Node.js features a built-in library that enables applications to function as a web server without the need for additional software like Apache HTTP Server, Nginx, or IIS. It employs event-driven, non-blocking I/O mechanisms, optimizing application performance and ensuring high extensibility. Asynchronous functions are integral to Node.js, enabling background processing for efficient task execution.

Notably, Node.js is preferred for products with high traffic, efficiently handling applications requiring rapid scaling, technological innovation, or expedited development of startup projects.

Applications using NodeJS:
- WebSocket server
- Notification system
- Applications that need to upload files on the client.
- Other real-time data applications

NodeJS Pros:

1. Efficient Single Thread Handling: Node.js stands out by efficiently managing numerous connections with just a single thread. This eliminates the need to create new threads for each query, minimizing RAM usage and ensuring rapid execution. Its non-blocking I/O using JavaScript contributes to low latency.

2. Support for JSON APIs: Node.js is well-suited for JSON web services due to its event-driven, non-blocking I/O architecture and compatibility with JavaScript.

3. Compatibility with Single Page Applications (SPAs): Node.js excels in handling concurrent requests and quick returns, making it ideal for single-page applications. It is particularly effective for applications where page reloading is not required, offering a seamless experience for users who generate a high volume of requests.

4. Unix Shell Tool Integration: Node.js seamlessly integrates with Unix shell tools, allowing the handling of multiple processes for optimal performance. It is frequently utilized by programmers to develop real web applications such as chat systems and feeds.

5. Streaming Data Capability: Node.js excels in handling multiple HTTP requests and responses, making it suitable for applications that involve streaming data. It can also build proxies to enhance data streaming efficiency.

6. Real-time Web Application Development: Node.js is particularly well-suited for developing real-time innovations, including chat applications and social networking services like Facebook and Twitter. Its adaptability extends to mobile applications, making it a preferred choice for real-time features.

NodeJS Cons:

1. Resource-Intensive Applications: Node.js, being written in C++ and JavaScript, may not be the best choice for resource-intensive applications that involve heavy file conversion, video encoding, or decoding. In such cases, developers should exercise caution and consider alternative technologies that are better suited to handle these computationally intensive tasks.

2. Specialized Purpose: While Node.js excels in developing web applications, its primary purpose aligns with other programming languages like Ruby, PHP, .NET, and Python. It is not designed to outperform all other languages in every context. Developers should set realistic expectations and choose the right tool for the job based on the specific requirements

of the project.

Despite these limitations, Node.js remains a powerful and widely used platform for building scalable and efficient web applications, particularly when dealing with asynchronous tasks and real-time applications. It's crucial for developers to assess the nature of their projects and select the most suitable technology stack for optimal results.

Node.js may not be the most suitable choice in certain scenarios:

1. Resource-Intensive Applications: Avoid using Node.js for applications that heavily rely on resource-intensive tasks, such as a video converter. Node.js may encounter bottlenecks when dealing with large files.

2. All-CRUD-Only Application: If your application primarily involves CRUD operations and heavy I/O tasks, Node.js may not be as performant as alternatives like PHP, which has optimized CRUD tasks over an extended period. Node.js, in such cases, may introduce unconventional APIs and not be the best fit.

3. Stability Concerns: Node.js has evolved over its 11 years of development, with the current version being v14.2.0. However, developers should be aware that APIs may change, and updates might not be backward compatible, potentially impacting the stability of applications.

4. Lack of Knowledge: Using Node.js without a sufficient understanding of its non-blocking/async nature can lead to challenges. The community's strength and support are crucial, and a lack thereof may result in difficulties for developers.

On the other hand, Node.js is well-suited for

1. Building RESTful APIs (JSON): Node.js excels in constructing RESTful APIs, particularly those dealing with JSON. It efficiently handles concurrent requests, making it suitable for high-volume scenarios.

2. Applications Requiring Alternative Connection Protocols: Node.js is versatile, supporting various connection protocols beyond HTTP. With TCP protocol backing, custom protocols can be easily implemented.

3. Real-Time Applications: Node.js is ideal for real-time applications where immediate and dynamic data updates are crucial.

4. Stateful Websites: Node.js efficiently handles stateful websites, simplifying caching processes. Persistent client states can be stored and shared among clients without relying on external memory, contributing to streamlined operations.

Express.js

Express.js, built on Node.js, enhances web and mobile development by supporting HTTP and middleware methods. It provides additional features that improve the programming environment without compromising Node.js speed. Express.js is a core component in popular Node.js frameworks like Sails.js and the MEAN stack (MongoDB, Express.js, AngularJS, Node.js)[14]. This underscores its versatility and widespread adoption in the Node.js development ecosystem.

MongoDB

MongoDB, an open-source database, stands as the leading NoSQL database, widely embraced by millions. It's crafted in one of today's most popular programming languages. Operating on the concepts of Collections and Documents, MongoDB ensures high performance, availability, and ease of scalability. It's cross-platform, offering versatility in deployment.

In the realm of NoSQL, MongoDB innovatively deploys a JavaScript Framework on JSON data types, deviating from Transact-SQL. This departure addresses the limitations of traditional RDBMS relational models, enhancing operational speed, functionality, and scalability.

With a focus on the Collection and Document approach, MongoDB continues to deliver efficient production, robust availability, and straightforward scalability. Its prominence stems from its ability to meet the evolving needs of modern data management.

ReactJS

Virtual-DOM

Virtual-DOM is a JavaScript object that holds information to recreate the DOM. It calculates changes between this object and the real DOM, optimizing the process of updating and rendering. It efficiently manages client data, enhancing performance by selectively updating only the changed parts of the DOM.

Component

React revolves around components rather than templates, distinguishing it from other frameworks. Components are created using the `createClass` function of the React object, serving as the entry point to the library. In React, HTML tags are generated using components, encapsulating them within layered objects for rendering.

Within React components, the `render` function holds paramount importance. This function is responsible for generating HTML tags and showcases the capability to process through Virtual DOM. Changes in data trigger immediate updates through Virtual DOM, ensuring efficient and responsive rendering.

Pros of ReactJS:

1. Efficient Data Updates: React excels in quickly updating data changes, thanks to its use of the Virtual DOM and efficient rendering mechanisms.

2. Libraries Flexibility: React is not a full-fledged framework, providing flexibility by allowing developers to choose libraries according to their needs, rather than imposing a specific set of tools.

3. JavaScript Familiarity: React is easily accessible for those familiar with JavaScript, lowering the learning curve for developers.[15]

Cons of ReactJS:

1. View-Centric: React primarily focuses on the View tier of the application, which can be seen as a limitation compared to complete frameworks like Angular.

2. Library Size: React's library size can be comparable to Angular's despite being just a library, leading to concerns about application size and performance.

3. Integration Challenges: Integrating ReactJS with common MVC frameworks may require reconfiguration, posing challenges for seamless adoption.

4. Steep Learning Curve for Beginners: ReactJS can be challenging for beginners in web development due to its unique concepts and the need to understand JSX, which might be unfamiliar to newcomers.[15]
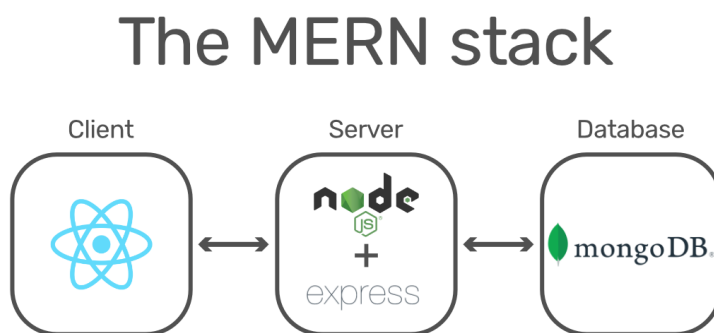
Figure 7.1 MERN Stack

CHAPTER-8

SYSTEM DESIGN & IMPLEMENTATION

System Design

Architectural Style

Microservices Architecture: Wirepool employs a microservices architecture, promoting modularity, scalability, and maintainability. Each service operates independently, facilitating continuous deployment and reducing system-wide failures.

Key Components

User Management Service: Responsible for user registration, authentication, profile management, and role-based access control. It integrates with authentication providers like OAuth and ensures data integrity via secure encryption mechanisms.

Service Request Service: This core component manages the lifecycle of service requests. It matches user requirements with professional skills, handles booking, scheduling, and ensures timely service delivery.

Payment Service: Integrates with payment gateways to facilitate secure transactions. It handles payment processing, invoicing, and ensures compliance with financial regulations.

Notification Service: Manages real-time notifications, sending alerts, updates, and reminders to users and professionals. It supports multiple communication channels, including SMS, email, and in-app notifications.

Review & Rating Service: Collects, manages, and displays user reviews and ratings. It employs sentiment analysis to filter content, ensuring authenticity and relevance.

Data Storage & Management

Database Architecture

User Database: Utilizes MongoDB, a NoSQL database, to store user profiles, authentication tokens, preferences, and activity logs.

Data Management Strategies

Data Partitioning: Implements sharding in MongoDB to distribute data across multiple servers, ensuring optimal performance and scalability.

Backup & Recovery: Deploys automated backup solutions and disaster recovery strategies to safeguard against data loss and ensure business continuity.

Security & Compliance

Authentication & Authorization

JWT: Implements JWT-based authentication for secure user sessions, token-based access control, and stateless communication.

OAuth 2.0: Integrates OAuth 2.0 for secure third-party application integrations, ensuring data privacy and user consent.

Hardware Details:

1.   **Smartphone Devices:**

The devices that are used in the development of the project/application are any Android or IOS devices running on Android 10 and above for the Android devices, and IOS 12   and above for the IOS devices.

2.   **Server Hardware**:

- Node.js: For server-side JavaScript runtime.
- Express.js: As a web application framework for building RESTful APIs.

Software Details:
1. **Frontend Development:**
   ● React Native Framework: Detail the use of React Native for building the Wirepool mobile application.
   ● Discuss how React Native was utilized, its advantages in cross-platform development, and the specific features or components developed using this framework.

2. **Backend Services:**

   MongoDB Database:

   ● **Purpose:** MongoDB is utilized as the primary database management system for storing essential user data, service details data and authentication details within the Wirepool application.

   ● **User Authentication Data:** MongoDB stores user-specific data, including Login credentials, User profiles, Service details, , and authentication tokens.

   ● **Security Measures:** Discuss the security measures implemented in MongoDB to safeguard user authentication data, such as encryption, hashing algorithms, access controls, etc.

   ● **Scalability and Performance:** Describe how MongoDB's scalability and performance features are utilized to handle user data efficiently, ensuring quick access and retrieval of authentication details.

   ● **Server-Side Technologies**: If there are server-side components or APIs, mention the technologies used.
   ● For instance, Node.js/Express.js for API development or any other backend services used in conjunction with MongoDB.

3. **Development Tools:**

   ● IDEs: Integrated Development Environments used for coding : Visual Studio Code.
   ● Version Control: The version control systems like Git, GitHub are utilized for collaboration and code management.
   ● Other Tools and Libraries: Mention any additional tools, libraries, or third-party dependencies essential to the Wirepool project development.

4. **Deployment and Hosting** :
   ● Explain the process of deploying the Wirepool mobile application to devices or app stores.
   ● If hosting is required, describe the hosting environment or cloud services used for deploying backend components or databases.

USE CASE DIAGRAM
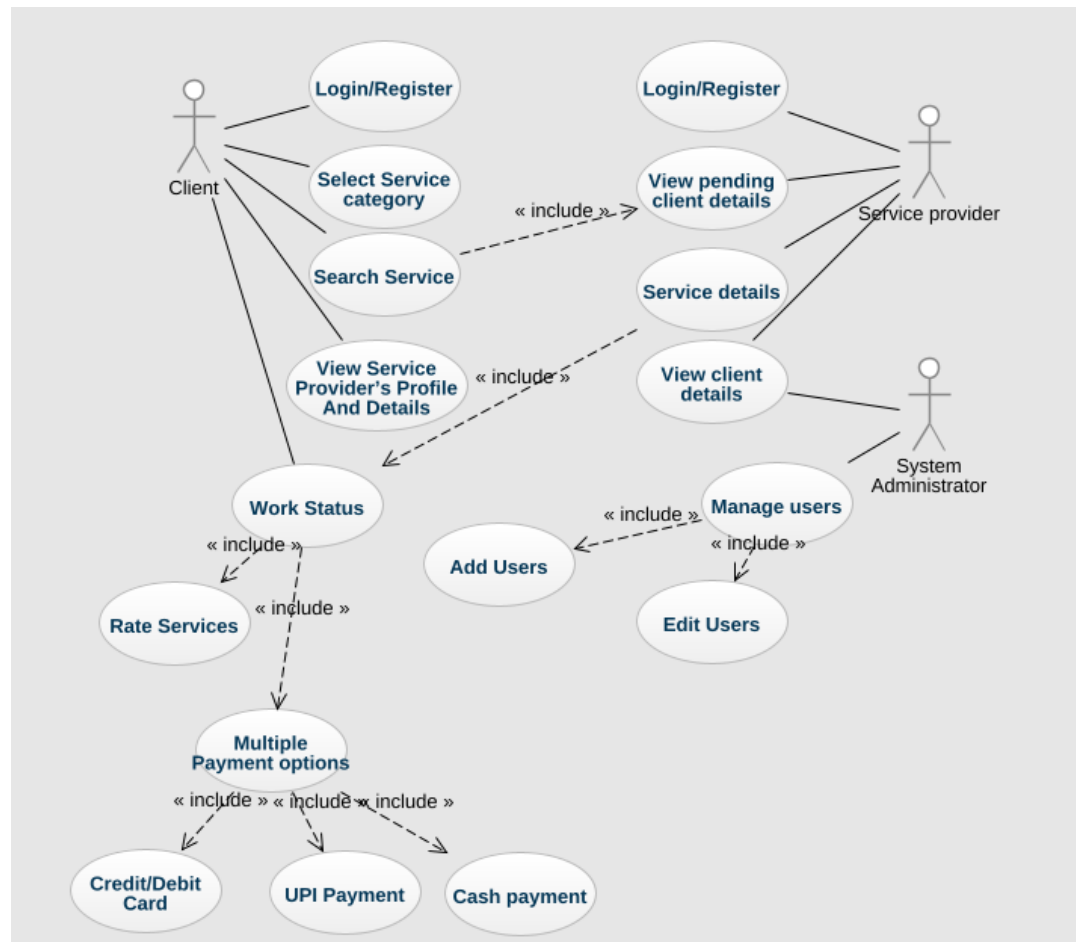
Figure 8.1 Use Case Diagram

1. **CLASS DIAGRAM :**

Creating a class diagram for an e-commerce MERN (MongoDB, Express.js, React.js, Node.js) project involves identifying the key classes and their relationships. Below is a simplified representation of the class diagram for the specified features: login/signup, home, search, bookings, and cart pages.
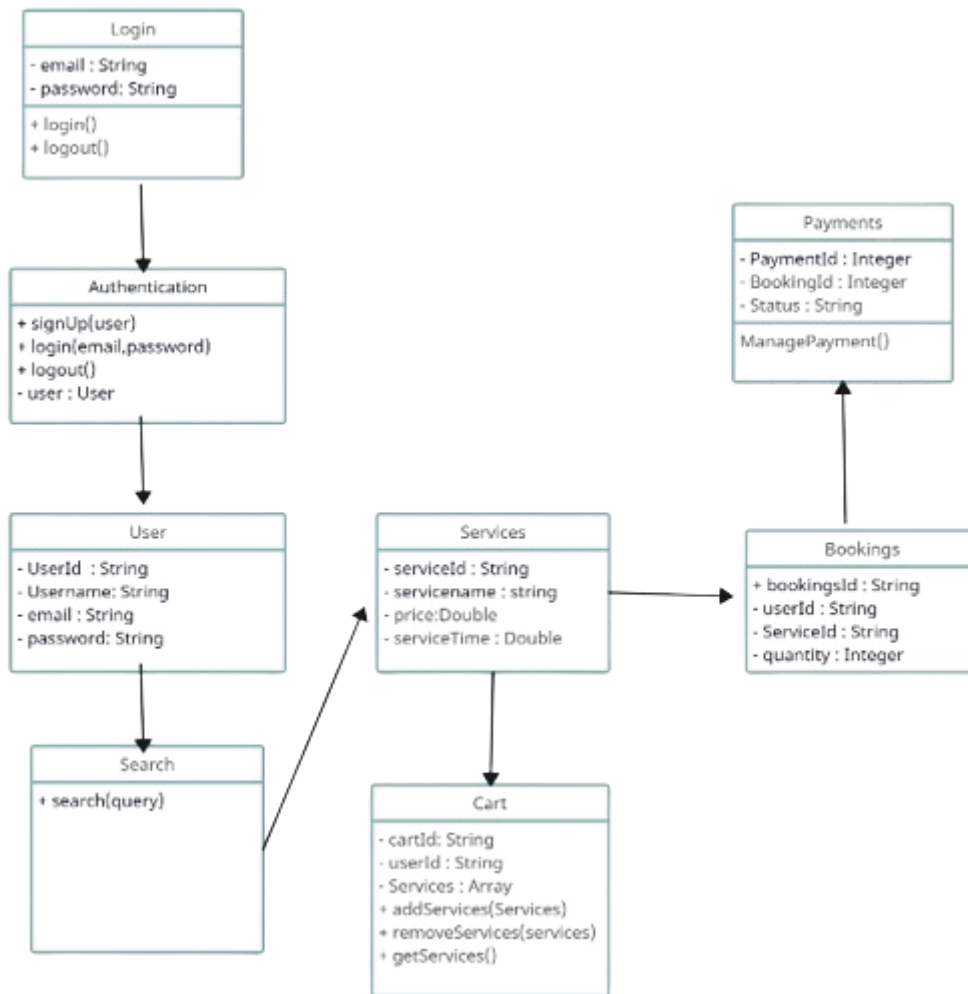
Figure 8.2 Class diagram

Explanation:

1. User Class:

  - Represents the users of the system.

  - Contains basic user information such as userId, username, email, and password.

2.Service Class:

  - Represents the services available for purchase.

  - Contains details like ServicetId, name, price, and time.

3. Booking Class:

  - Represents the booking information.

  - Includes bookingId, userId, serviceId, and time.

4. Auth Class:

  - Manages user authentication.

  - Includes methods for user registration, login, and logout.

5. Cart Class:

  - Represents the shopping cart.

   - Includes cartId, userId, and an array of items. Provides methods to add and remove items from the cart.

6. Search Class:
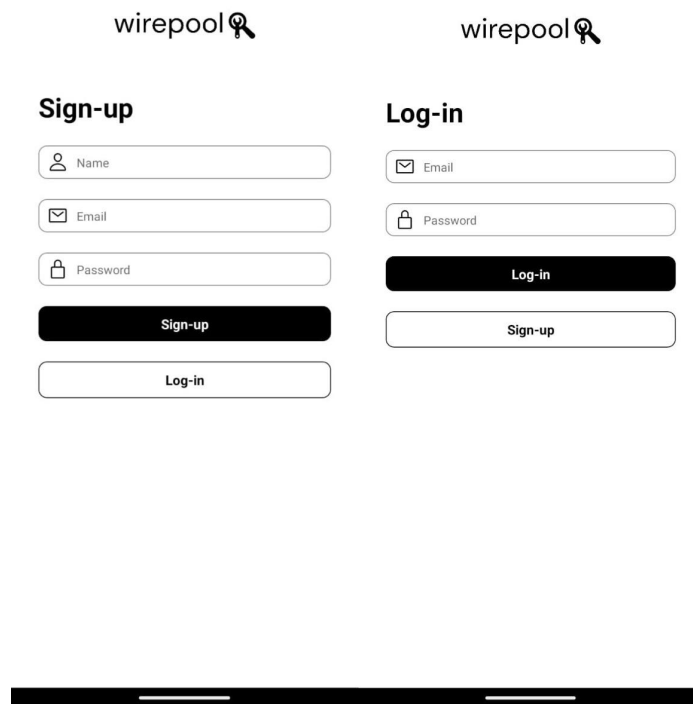
  - Handles the searching functionality.

  - Includes a method to search for products based on a query.

These classes represent the major entities and functionalities in an e-commerce MERN project. Depending on the specific requirements and features of your project, you may need to extend or modify the class diagram. Additionally, consider the use of associations, such as aggregation or composition, to represent relationships between classes.

FRONTEND

Pages of wirepool

- **Login / Register page :**

  This page serves as the entry point for users to access your app. New users can register for an account, providing necessary details, while returning users can log in. It's a crucial component for user authentication and access to personalized features.

Figure 8.3 Login/Signup Page

- **Home page :**

The home page is the main screen users see upon logging in. It provides a personalized and dynamic view, showcasing relevant content, recommendations, or featured items. The goal is to engage users and encourage exploration of the app's offerings.

Figure 8.4 Home Page

- **Search page :**

  The search page allows users to discover specific content or products by entering search queries. It typically includes a search bar and filters to refine results. Users can easily navigate through the app's offerings, enhancing their overall experience.

- **Cart page :**

  The cart page displays the items users have added to their shopping cart or booking list. Users can review, modify, or proceed to checkout/book directly from this page. It's a crucial step in the e-commerce or booking process, providing a clear overview of selected items.
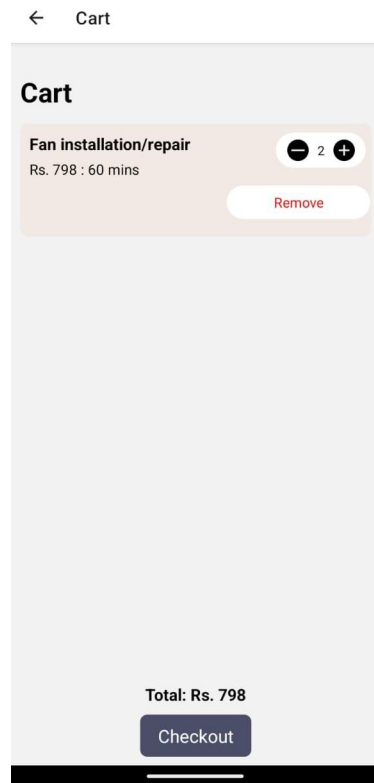
Figure 8.5 Cart Page

● **Bookings page :**

The bookings page is relevant for apps that involve reservations, appointments, or any form of scheduled events. Users can view, modify, or cancel their existing bookings. This page helps users manage their commitments within the app and provides a seamless booking experience.
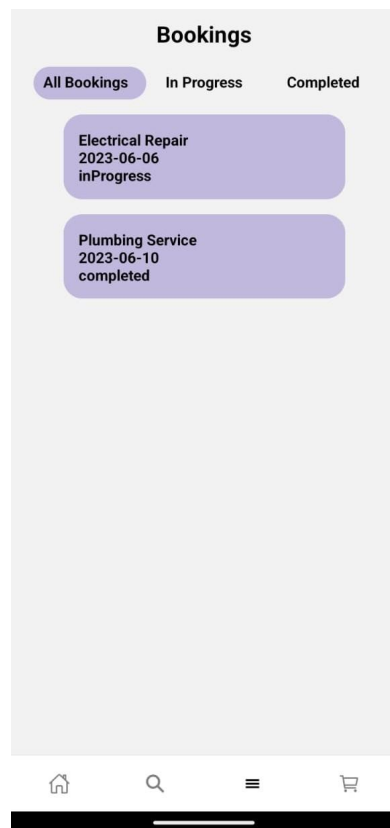
Figure 8.6 Bookings Page

BACKEND
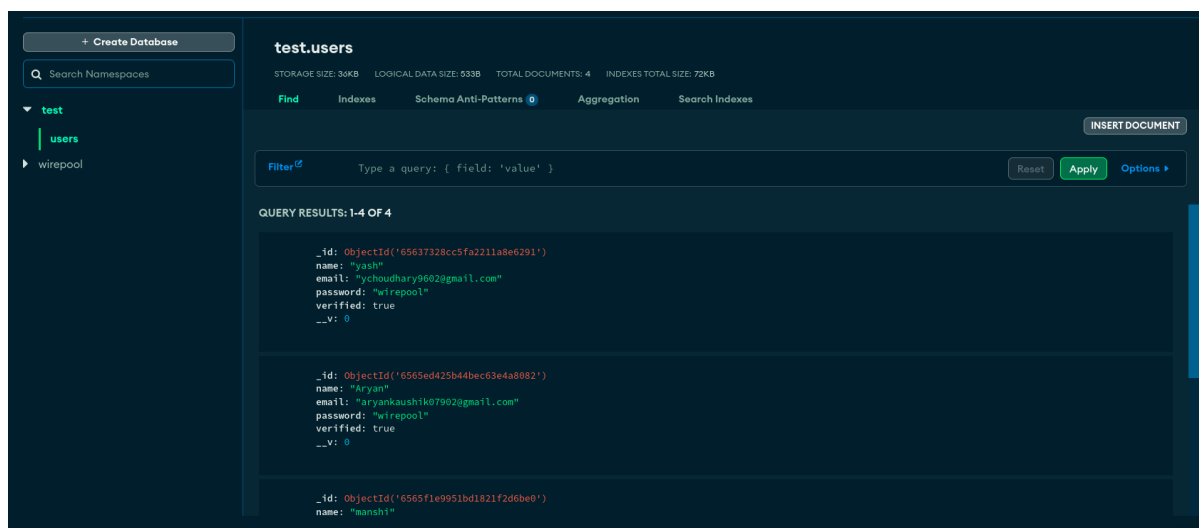
Figure 8.7 MongoDB Atlas

Figure 8.8 Running Database

Figure 8.9 Packages

CHAPTER-9

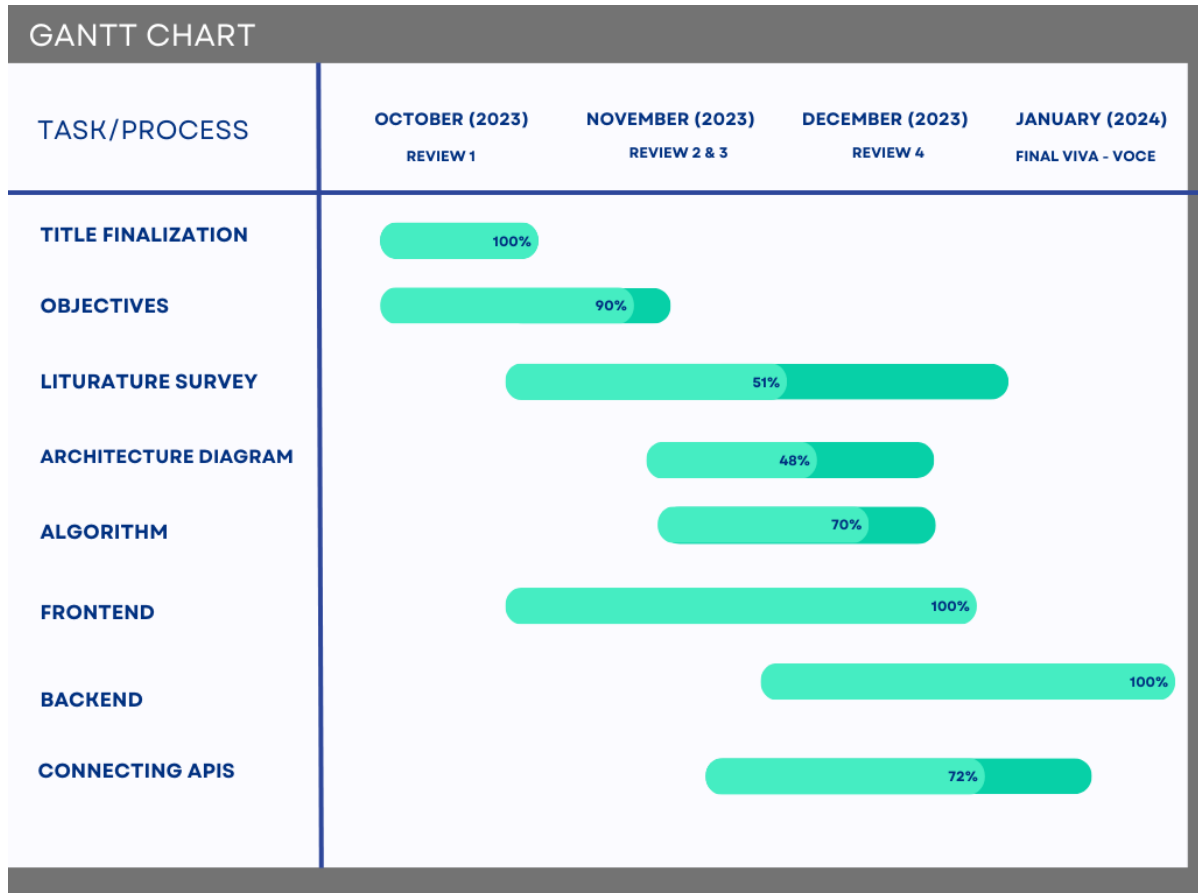# TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)

Figure 9.1 Gantt Chart

CHAPTER-10

RESULTS AND DISCUSSIONS

1. User Interface (UI) and User Experience (UX):
- The user interface was designed to be intuitive, ensuring ease of navigation for both end users and professionals.
- Feedback from beta testing highlighted positive responses to the user-friendly design, contributing to a positive overall user experience.
- Streamlined service request processes and clear communication channels improved user satisfaction.

2. Authentication and Security:
- Robust authentication mechanisms were implemented to ensure the security of user data and transactions.
- Encryption protocols and secure login methods provided a reliable and trustworthy platform for both end users and professionals.
- Continuous monitoring and updates were undertaken to address emerging security concerns, ensuring a secure environment for all stakeholders.

3. Service Request Management:
- The platform successfully streamlined service requests, allowing users to submit and track their requests efficiently.
- Real-time updates and notifications kept users and professionals informed about the status of ongoing projects, enhancing communication and transparency.
- Feedback mechanisms were integrated to gather insights and improve the service request process continuously.

4. Collaboration and Communication:
- Wirepool facilitated seamless communication between end users and professionals, fostering collaboration.
- Instant messaging, video conferencing, and file sharing features contributed to effective communication, leading to successful service deliveries.
- Positive feedback was received regarding the platform's ability to bridge the gap between users and professionals, creating a collaborative environment.

5. Diverse Service Offerings:
- The platform successfully accommodated a diverse array of services, including connections with Original Equipment Manufacturers (OEMs), Process Specialists, Electricians, and Freelance Consultants.
- The variety of services contributed to the platform's versatility, attracting a wide range of users seeking different expertise.

6. Future Vision and Ecosystem Impact:
- Wirepool's vision of harmonizing expertise and demand was evident in the positive

impact on the service delivery ecosystem.

   - The platform has the potential to revolutionize how services are accessed, delivered, and experienced, creating a thriving and interconnected ecosystem.

   - Ongoing improvements and updates are planned to enhance the platform's capabilities and adapt to evolving industry needs.

CHAPTER-11

CONCLUSION

The journey of conceptualizing, developing, and deploying Wirepool has been both challenging and rewarding, culminating in a platform that bridges the gap between users and specialized professionals in an unprecedented manner. This project, built on the MERN stack, signifies the power of innovative technology solutions in addressing real-world challenges and enhancing user experiences.

Key Takeaways and Reflections:

Innovation in Service Delivery: Wirepool's success lies in its ability to innovate the traditional service delivery model. By offering an integrated platform that facilitates seamless interactions between users and professionals, Wirepool has redefined the standards of service accessibility and quality.

User-Centric Design: The emphasis on an intuitive and user-friendly interface has been instrumental in driving user adoption and engagement. Wirepool's commitment to understanding user needs and preferences has resulted in a platform that resonates with its target audience.

Collaborative Ecosystem: Wirepool's ecosystem thrives on collaboration. The diverse array of professionals, ranging from OEMs to Freelance Consultants, reflects the platform's inclusive approach and commitment to serving varied user requirements.

Ethical Considerations: As with any digital platform, Wirepool acknowledges the ethical considerations inherent to the gig economy. The platform is committed to fostering fairness, transparency, and accountability in its operations, ensuring a conducive environment for both users and professionals.

Future Outlook: The potential for Wirepool's growth and expansion is immense. With a strong foundation, positive user feedback, and a clear roadmap for future enhancements, Wirepool is well-positioned to scale new heights and establish itself as a leading player in the service industry landscape.

In conclusion, Wirepool represents more than just a digital platform; it symbolizes a vision for a connected, efficient, and user-centric service ecosystem. As we reflect on the accomplishments and milestones achieved, we remain committed to continuous innovation, user empowerment, and creating value for all stakeholders involved. The Wirepool journey is ongoing, with the future holding promise for further growth, expansion, and transformative impact in the realm of service delivery and management.

CHAPTER – 12

REFERENCES

[1]E-commerce Definition – What is E-commerce? [Internet]. Shopify.com. Available from: https://www.shopify.com/encyclopedia/what-is-ecommerce

[2]Advantages of E-commerce [Internet]. Thebalancesmb.com 2019 [cited 20 November 2019]. Available from: https://www.thebalancesmb.com/ecommercepros-and-cons-1141609

[3]Dr. Santosh Kumar Shukla, Shivam Dubey, Tarun Rastogi and Nikita Srivastava. Application using MERN Stack. International Journal for Modern Trends in Science and Technology 2022, 8(06), pp. 102-105.
https://doi.org/10.46501/IJMTST0806014

[4]Vipul Kaushik, Kamali Gupta, Deepali Gupta - React Native Application Development
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3330011

[5]Qui Chong - Service supply and demand matching method and system

[6]Buddhini Gayathri Jayatilleke, Gaya R. Ranawaka, Chamali Wijesekera and Malinda C.B. Kumarasinha - Development of mobile application through design-based research
https://www.researchgate.net/publication/327663611_Development_of_mobile_application_through_design-based_research

[7]Thomas C. G. & A. Jayanthila Devi - A Study and Overview of the Mobile App Development Industry
https://www.researchgate.net/publication/352490326_A_Study_and_Overview_of_the_Mobile_App_Development_Industry

.[8]Laith T. Khrais, Abdullah M. Alghamdi - The Role of Mobile Application Acceptance in Shaping E-Customer Service
https://www.researchgate.net/publication/350205962_The_Role_of_Mobile_Application_Acceptance_in_Shaping_E-Customer_Service

[9]P.K. Paul & Sreeramana Aitha - MOBILE APPLICATIONS SECURITY
https://www.researchgate.net/publication/336845625_MOBILE_APPLICATIONS_SECURITY_AN_OVERVIEW_AND_CURRENT_TREND

[10]Dan Johansson, Karl Andersson - Mobile e-Services: State of the Art, Focus Areas, and Future Directions
https://www.researchgate.net/publication/270790245_Mobile_e-Services_State_of_the_Art_Focus_Areas_and_Future_Directions

[11]Md. Abdur Razzaque, Amitava Chatterjee - Mobile computing and its applications in healthcare

[12]JavaScript [Internet]. Mozilla.org. Available from:
https://developer.mozilla.org/en-US/docs/Web/JavaScript

[13]NodeJS Introduction [Internet]. Tutorialspoint.com. Available from:
https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm

[14]Express.js Introduction [Internet]. Mozilla.org. Available from:
https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction

[15]Pros and Cons of ReactJS [Internet]. Javatpoint.com. Available from:
https://www.javatpoint.com/pros-and-cons-of-react

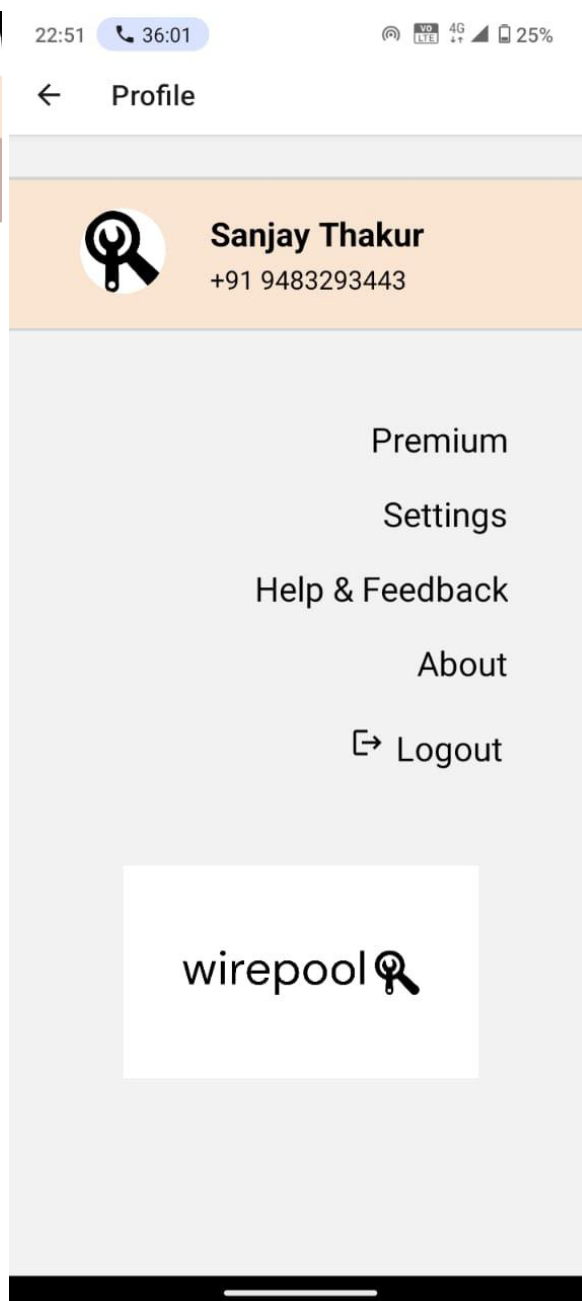# APPENDIX-A

PSUEDOCODE
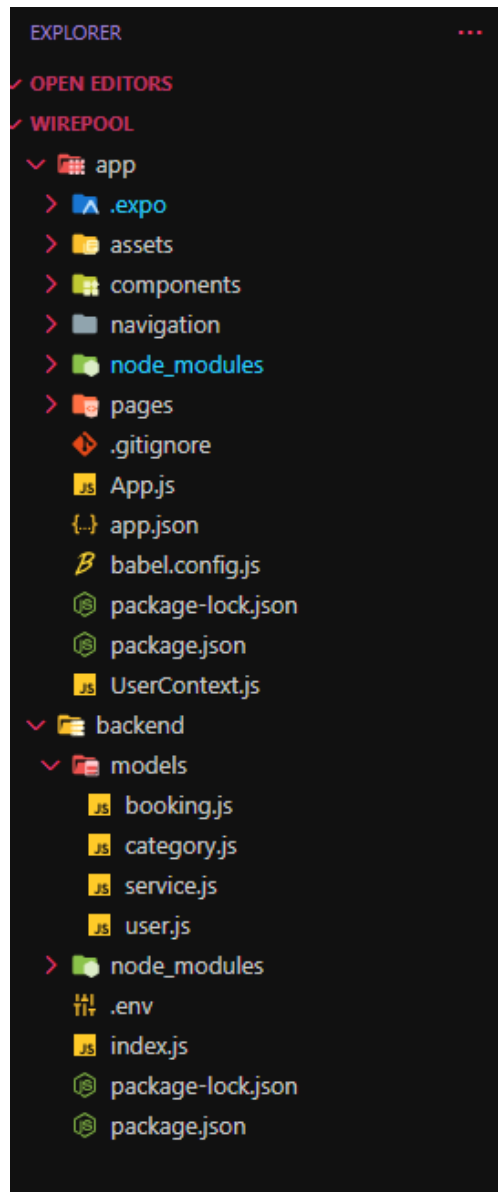
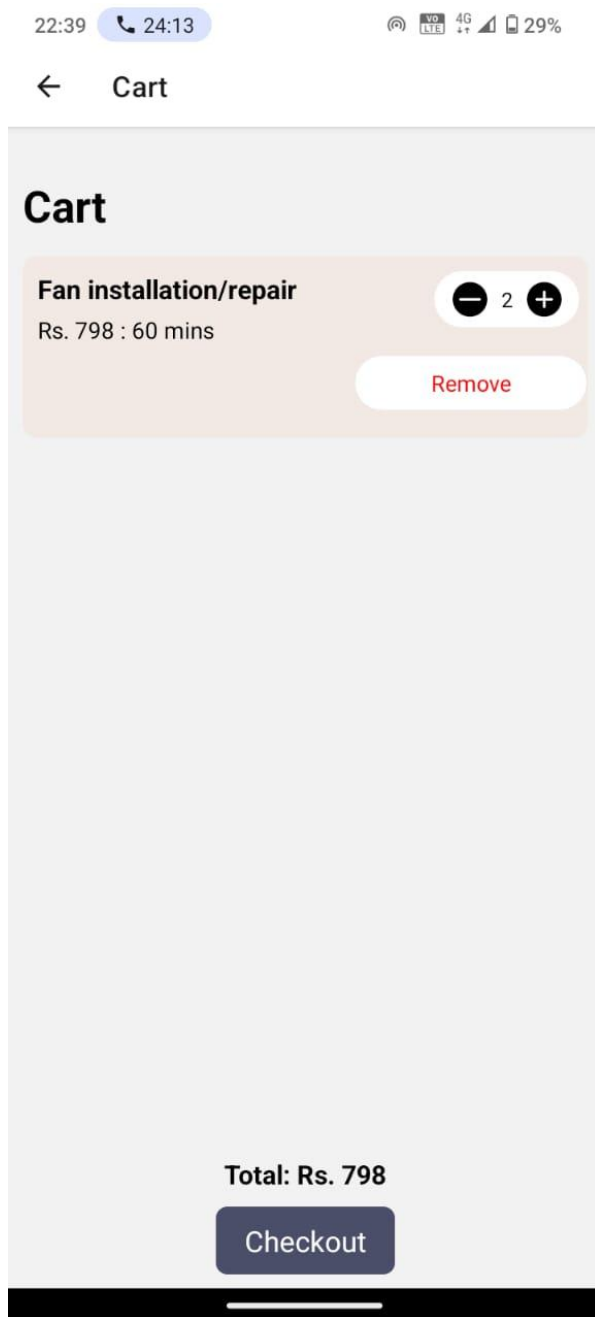File Structure

APPENDIX-B
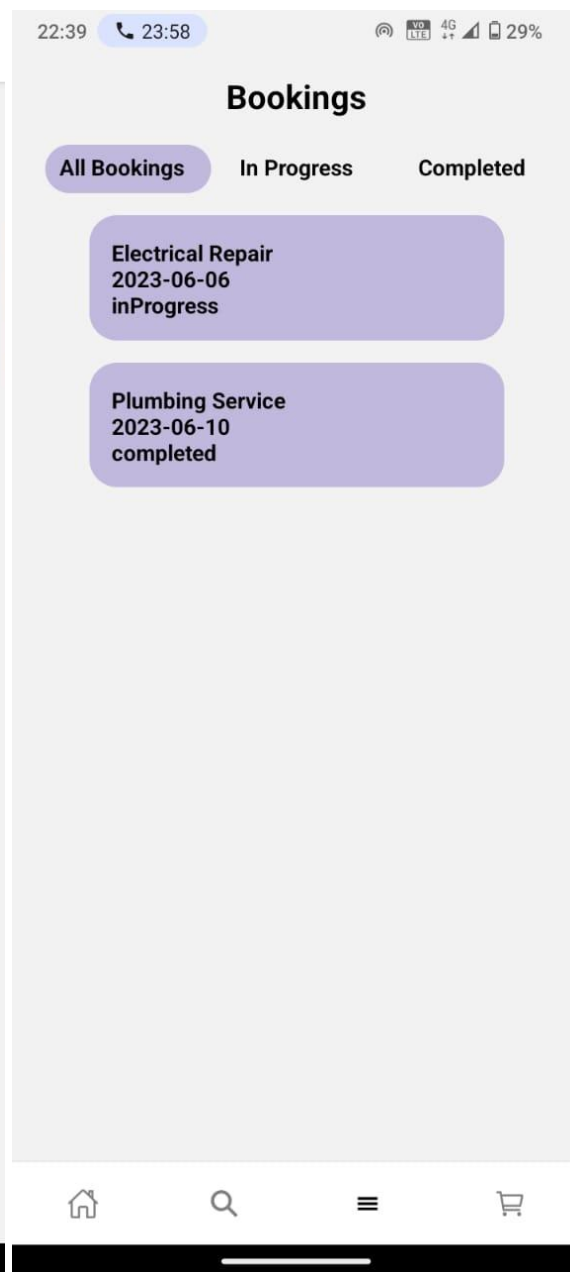# SCREENSHOTS

Home Page with

all category of services

Profile Page

Cart Component   Bookings Page

APPENDIX-C
ENCLOSURES

1. Conference Paper Presented Certificates of all students.

2. Include certificate(s) of any Achievement/Award won in any project related event.

3. Similarity Index / Plagiarism Check report clearly showing the Percentage (%). No need of page-wise explanation.