

SHERISES: A Web-Based Women Empowerment Platform with Real-Time Service Tracking

Jakkula Jayanthi¹, Kothakapu Kavya², Kodavatikanti Sathwika³

¹²³Department of Computer Science and Engineering, JNTUH University College of Engineering, Hyderabad, Telangana, India

Abstract - Women often use multiple digital platforms for jobs, transportation, healthcare, and other daily services, making the process inconvenient and less secure. **SHERISES** is a web-based women empowerment platform that integrates these services into a single application, allowing users to create profiles, upload resumes, apply for jobs, book trusted services, and track bookings in real time. The platform is developed using **React.js**, **Node.js**, **Express.js**, and **MongoDB**, with **Socket.IO** for live updates, **React Leaflet** with **OpenStreetMap** for location tracking, and **JWT** with **bcrypt.js** for secure authentication. By combining employment, service booking, and safety features, SHERISES provides a secure, scalable, and user-friendly solution that supports women's empowerment and can be extended with AI recommendations, SOS support, digital payments, and multilingual features.

Key Words: Women Empowerment, Web Application, React.js, Node.js, MongoDB, Socket.IO, Live Tracking, Service Booking, Job Portal, OpenStreetMap.

1. INTRODUCTION

Technology has transformed the way people search for jobs, access transportation, and book essential services through digital platforms. However, most existing applications provide only a single service, requiring users to switch between multiple apps for their daily needs, which reduces convenience and efficiency.

For women, this fragmented approach also raises concerns about safety, trust, and accessibility. Many platforms lack women-focused services, secure authentication, and real-time tracking, making it difficult to access reliable services through a single trusted platform.

To address these challenges, **SHERISES (She Rises)** is proposed as a centralized web-based women empowerment platform. It enables users to create profiles, upload resumes, search and apply for jobs, book women-centric services, and track bookings in real time through one integrated application.

The platform is implemented using **React.js** for the frontend, **Node.js** and **Express.js** for the backend, and **MongoDB** for data storage. **Socket.IO** supports real-time communication, while **React Leaflet** with **OpenStreetMap** provides live location tracking. Security is ensured using **JWT authentication** and **bcrypt.js** password encryption.

The main objective of SHERISES is to deliver a secure, user-friendly, and scalable platform that promotes women's employment, safety, and accessibility. By integrating multiple

services into a single system, it offers a convenient digital solution and provides a foundation for future enhancements such as AI-based recommendations, SOS support, and digital payments.

Novelty of the Proposed Platform: Unlike existing applications that focus on only one domain, such as employment portals, transportation services, or household service platforms, SHERISES combines women-centric employment opportunities, trusted service booking, secure user authentication, resume management, profile management, and real-time service tracking within a single web application. This integrated approach minimizes application switching, enhances user convenience, improves safety through live tracking, and provides a centralized digital platform tailored specifically for women's empowerment.

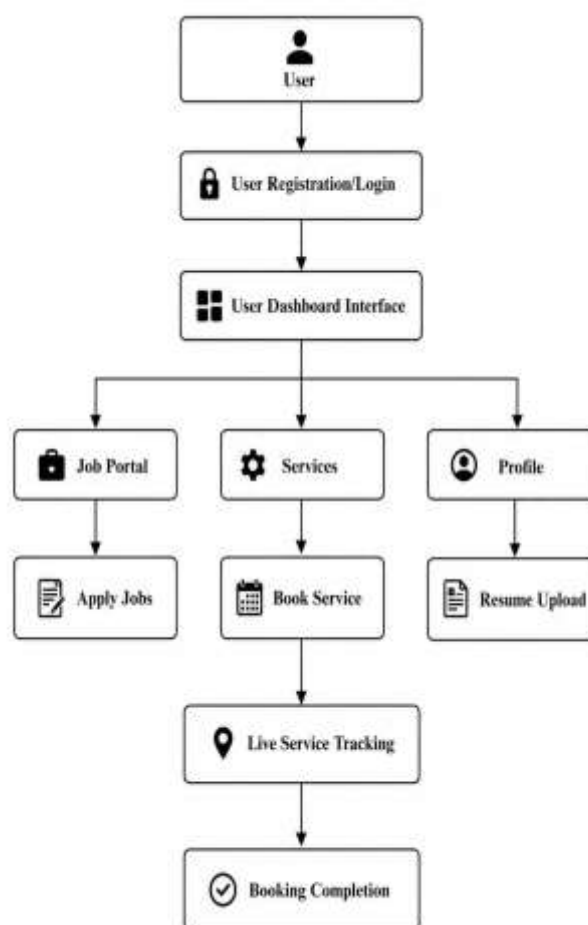


Fig. 1. Overall Workflow of SHERISES Platform

2. PROBLEM STATEMENT

Despite the availability of numerous online platforms, women still face several challenges while accessing employment opportunities and essential services. Existing applications generally focus on a single domain, such as job recruitment, transportation, healthcare, or home services. This forces users to switch between multiple platforms to complete related tasks, making the overall experience less efficient and more time-consuming.

Another major limitation is the lack of women-centric platforms that prioritize safety and trust. Most service applications do not provide features such as integrated profile management, centralized booking history, or continuous live tracking of service providers. In many cases, users have limited visibility regarding the current location or status of a booked service, which can create uncertainty and reduce confidence in the platform.

Similarly, employment portals provide facilities for searching and applying for jobs but rarely integrate service-related features. Women seeking both employment opportunities and trusted services must therefore depend on separate applications, increasing complexity and reducing overall convenience.

The absence of a unified platform results in fragmented information, repeated user registrations, inconsistent user experiences, and limited security features. These shortcomings highlight the need for a centralized solution that combines employment, service booking, profile management, and real-time tracking while ensuring user safety.

The SHERISES platform has been designed to overcome these limitations by providing a secure, integrated, and women-focused web application that brings together multiple services under one platform.

Table -1: Limitations of Existing Systems

Existing Platform	Primary Function	Limitations
LinkedIn	Professional networking and jobs	No service booking or live tracking
Naukri	Job search	No integrated transportation or household services
Uber	Transportation	No employment opportunities or profile management
Rapido	Bike taxi service	Limited women-focused safety features

Urban Company	Home services	No employment portal or resume management
SHERISES	Integrated women empowerment platform	Combines jobs, services, tracking, and profile management

3. OBJECTIVES

The primary objective of the SHERISES platform is to develop a secure and user-friendly web application that supports women by providing employment opportunities, trusted services, and real-time service tracking within a single platform. The project aims to eliminate the need for multiple independent applications by integrating essential services into one centralized system.

The specific objectives of the proposed system are as follows:

- To develop a centralized platform that combines job opportunities and women-centric services.
- To implement a secure registration and login mechanism using JWT authentication and encrypted passwords.
- To allow users to upload resumes and apply for suitable job opportunities directly through the platform.
- To offer women-specific services such as bike taxis, cab services, beautician services, babysitting, grocery delivery, food delivery, and healthcare consultation.
- To support live location tracking for booked services using Socket.IO, React Leaflet, and OpenStreetMap.
- To enable profile management features where users can update personal details and maintain their employment information.
- To design a responsive and user-friendly interface that can be accessed from different devices.
- To demonstrate the practical implementation of modern full-stack web technologies for solving real-world problems.
- To create a scalable platform that can support future enhancements such as AI recommendations, digital payments, and emergency safety features.

The overall objective is not only to build a functional application but also to improve accessibility, safety, and convenience for women through the effective use of modern web technologies.

4. LITERATURE SURVEY

With the rapid advancement of information technology, several online platforms have been developed to simplify employment, transportation, and household service management. Although these applications have significantly improved digital accessibility, most of them focus on a single functionality and do not provide a comprehensive solution that addresses multiple user requirements simultaneously.

Employment platforms such as **LinkedIn**, **Naukri**, and **Indeed** have become popular for connecting job seekers with recruiters. These applications offer resume management, job recommendations, and professional networking opportunities.

However, they are primarily limited to employment-related activities and do not include features such as service booking or real-time tracking.

Similarly, transportation platforms including **Uber**, **Rapido**, and **Ola** provide efficient ride-booking facilities. These applications have introduced GPS-based tracking, digital payments, and driver information to improve user experience. Despite these features, they are designed specifically for transportation and do not integrate employment services or profile management functionalities.

Service-oriented platforms such as **Urban Company** offers home maintenance and personal care services, including beauticians, electricians, and healthcare professionals. Although these platforms simplify household service management, they lack integrated employment modules and centralized user profiles that combine multiple services.

Several research studies have also proposed women safety applications that include emergency alert systems, GPS tracking, and SOS notifications. While these systems contribute significantly to women's safety, they generally focus only on emergency situations and do not provide employment opportunities or everyday service management.

From the analysis of existing applications and previous research, it becomes evident that there is no unified platform that combines employment opportunities, women-focused services, secure profile management, and live tracking within a single web application.

SHERISES has been developed to bridge this gap by integrating all these functionalities into one centralized platform. The application focuses on providing convenience, safety, transparency, and accessibility while utilizing modern web development technologies.

Previous Research

Gupta and Sharma proposed a GPS-based women safety system using IoT that improves emergency response.

Gondaliya and Rathod developed an Android-based women safety application with emergency alert features.

Chavan and Deshmukh presented a GPS-GSM based women safety system.

Table -2: Literature Comparison

System	Jobs	Service Booking	Live Tracking	Women - Centric	Resume Upload
LinkedIn	✓	✗	✗	✗	✓
Naukri	✓	✗	✗	✗	✓
Uber	✗	Transportation Only	✓	Partial	✗

Rapido	✗	Bike Taxi	✓	Partial	✗
Urban Company	✗	Home Services	Limited	Partial	✗
SHERISES	✓	✓	✓	✓	✓

5. EXISTING SYSTEM

At present, users depend on different online platforms to perform different activities. Employment opportunities are searched through job portals, transportation services are accessed using cab booking applications, and home services are booked through separate service platforms. Although each application performs its individual task efficiently, users are required to switch between multiple applications to complete their daily activities.

Most existing systems have the following limitations:

- Services are distributed across multiple independent platforms.
- User information must be maintained separately in each application.
- No centralized booking history is available.
- Limited integration between employment and service modules.
- Women-specific safety features are minimal.
- Real-time tracking is available only for transportation services.
- Repeated registrations increase user effort.
- Overall user experience becomes fragmented.

Because of these limitations, users spend additional time managing different accounts and applications instead of accessing services through a unified platform.

Limitations of Existing System

- Lack of integrated platform.
- Separate applications for jobs and services.
- No centralized profile management.
- Limited women-focused features.
- Reduced transparency during service delivery.
- Inconsistent user experience.

6. PROPOSED SYSTEM

The proposed SHERISES platform addresses the shortcomings of existing systems by integrating employment services, women-centric service booking, profile management, and live tracking into one centralized web application.

The platform is developed using a client-server architecture where the React.js frontend communicates with Node.js and Express.js backend services through REST APIs. MongoDB stores application data including users, jobs, bookings, and services. Socket.IO enables real-time communication for live

location tracking, while React Leaflet integrated with OpenStreetMap provides interactive map visualization.

The application enables users to perform all major activities from a single dashboard. After successful registration and login, users can browse available jobs, upload resumes, save interesting job opportunities, apply for jobs, book services, monitor ongoing bookings, and update their personal profiles.

Unlike existing systems, SHERISES focuses specifically on women users by offering trusted service providers and centralized management of all activities.

The primary novelty of SHERISES lies in its ability to integrate multiple women-oriented services within a single web platform instead of relying on separate applications. The system combines employment opportunities, trusted service booking, centralized profile management, resume handling, secure JWT-based authentication, and Socket.IO-enabled real-time tracking into one unified solution. This integration enhances transparency, reduces user effort, improves service accessibility, and provides a more secure and convenient digital experience for women.

The modular architecture facilitates future expansion and allows additional features to be integrated without affecting existing functionalities.

Major Features of Proposed System

User Module

The user module supports secure registration, authentication, dashboard access, profile management, and personalized services.

Job Management Module

Users are able to search available jobs, view job descriptions, upload resumes, save jobs, and submit applications.

Service Booking Module

The platform offers multiple women-centric services including transportation, beautician services, babysitting, grocery delivery, healthcare consultation, and food delivery.

Live Tracking Module

Users can conveniently monitor the real-time location of booked service providers through an interactive map interface.

Profile Management Module

Users can efficiently update personal information, upload resumes, and maintain booking history.

Admin Module

Administrators can monitor bookings, manage available services, update booking status, and maintain platform operations.

7. ADVANTAGES OF PROPOSED SYSTEM

The SHERISES platform provides several advantages over traditional standalone applications.

- Centralized management of jobs and services.
- Improved convenience through single-platform access.
- Women-focused design for better safety and trust.
- Secure authentication using JWT and encrypted passwords.
- Real-time communication using Socket.IO.
- Live location tracking using OpenStreetMap.
- Resume management and job applications in one platform.
- Easy scalability for future enhancements.
- Responsive design compatible with desktops and mobile devices.
- Reduced dependency on multiple independent applications.

Table -3: Existing System vs Proposed System

Feature	Existing System	Proposed SHERISES System
Centralized Platform	X	✓
Job Portal	Partial	✓
Women-Centric Services	X	✓
Live Tracking	Limited	✓
Resume Upload	Partial	✓
Secure Authentication	Partial	✓
Booking History	Limited	✓
Real-Time Updates	Partial	✓

Integrated Dashboard	X	✓
----------------------	---	---

8. SYSTEM ARCHITECTURE

The SHERISES platform follows a layered client-server architecture to ensure modularity, scalability, and maintainability. Each layer is responsible for a specific set of operations, allowing the application to function efficiently while keeping different components independent.

The Presentation Layer consists of the React.js frontend, which provides users with an interactive and responsive interface. Users interact with the system through pages such as Login, Dashboard, Jobs, Services, Tracking, and Profile Management.

The Application Layer is implemented using Node.js and Express.js. It receives requests from the frontend, validates user inputs, performs business logic, authenticates users using JWT, and communicates with the database.

The Data Layer uses MongoDB to store user information, job details, service bookings, applications, and tracking information. Mongoose is used as an Object Data Modeling (ODM) library to simplify database interactions.

The Real-Time Communication Layer is implemented using Socket.IO, enabling instant communication between users and service providers. This ensures that location updates are reflected immediately on the user's screen without requiring manual refreshes.

Finally, the Mapping Layer integrates React Leaflet with OpenStreetMap to display interactive maps and live service provider locations.

This layered architecture ensures high performance, better maintainability, and easier scalability for future development.



Fig. 2. Layered Client-Server Architecture of SHERISES

9. REQUIREMENT ANALYSIS

Requirement analysis is one of the most important phases in software development because it helps identify what the system should accomplish and the constraints under which it must operate. A clear understanding of user requirements allows developers to build a system that meets user expectations while remaining secure, scalable, and easy to maintain.

For the SHERISES platform, the requirements were gathered by studying existing job portals, transportation applications, service booking platforms, and women safety applications. The analysis revealed that users prefer a centralized platform that reduces the need to switch between multiple applications. Based on these observations, the functional and non-functional requirements of the proposed system were identified.

The requirements were categorized into two major groups:

- Functional Requirements
- Non-Functional Requirements

These requirements served as the foundation for designing the application architecture, database, user interface, and backend services.

9.1 Functional Requirements

Functional requirements define the operations that the system must perform to satisfy user needs. These requirements define the core functionalities available to users and administrators.

Table -4: Functional Requirements

Require ment ID	Functional Requirement	Description
FR1	User Registration	Allows new users to create an account.
FR2	User Login	Authenticates users using secure credentials.
FR3	Resume Upload	Enables users to upload resumes for job applications.
FR4	Job Search	Displays available job opportunities.
FR5	Job Application	Allows users to apply for suitable jobs.
FR6	Save Jobs	Stores preferred job listings for future reference.
FR7	Service Booking	Enables booking of women-centric services.

FR8	Live Tracking	Displays real-time location of service providers.
FR9	Booking History	Allows users to view previous bookings.
FR10	Profile Management	Enables updating personal information.
FR11	Admin Service Management	Allows administrators to manage services.
FR12	Booking Status Update	Allows administrators to update booking progress.

Explanation

The functional requirements ensure that users can complete all major activities without leaving the platform. From account creation to booking services and tracking them in real time, each module is designed to operate independently while communicating with other modules through REST APIs.

9.2 Non-Functional Requirements

Non-functional requirements define the quality attributes of the system. These requirements ensure that the application performs efficiently under different operating conditions.

Table -5: Non-Functional Requirements

Requirement Type	Description
Security	User passwords must be encrypted using bcrypt and authentication should use JWT tokens.
Performance	System responses should be fast with minimal delay during page loading and API requests.
Reliability	Services should remain available without unexpected failures.
Scalability	The architecture should support future feature additions.

Availability	The application should remain accessible whenever required.
Usability	The interface should be simple and easy to navigate.
Maintainability	Individual modules should be easy to modify without affecting the entire application.
Compatibility	The application should run on major web browsers.

Explanation

These requirements enhance the overall quality of the application. Security mechanisms protect user data, while scalability allows future enhancements such as mobile applications, payment gateways, and AI-based recommendations. A responsive interface and modular architecture ensure that the platform remains user-friendly and maintainable.

10. UML DIAGRAMS

Unified Modeling Language (UML) diagrams provide a graphical representation of the software system before implementation. They help developers understand system behavior, relationships between components, and interactions among users and modules.

The SHERISES platform uses four UML diagrams:

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Deployment Diagram

These diagrams simplify system analysis and improve software documentation.

10.1 Use Case Diagram

The Use Case Diagram illustrates the interaction between external users and the SHERISES platform. It identifies the major functions available to both users and administrators.

A normal user can perform activities such as registration, login, uploading resumes, searching jobs, applying for jobs, booking services, tracking service providers, cancelling bookings, and viewing booking history. The administrator manages services, monitors bookings, and updates booking statuses.

This diagram illustrates a high-level overview of the system and helps identify the responsibilities of each actor.

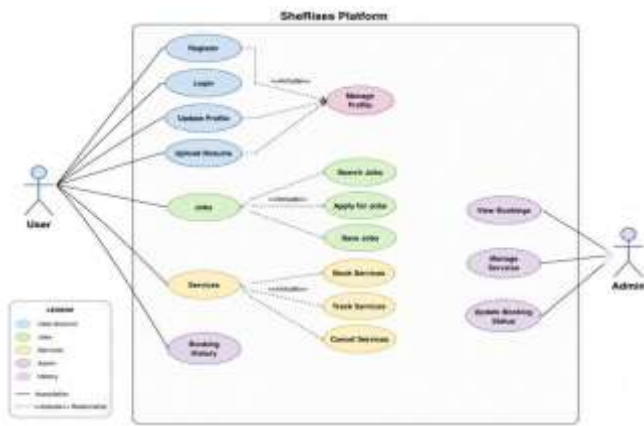


Fig. 3. Use Case Diagram of SHERISES

10.2 Class Diagram

The Class Diagram depicts the static structure of the application. It shows the classes used in the system along with their attributes, methods, and relationships.

The primary classes include:

- User
- Job
- Application
- Service
- Booking
- Tracking

The User class stores registration details and authentication information. The Job class contains job descriptions and company information. The Application class manages job applications submitted by users. The Service class stores available services, while the Booking class maintains booking records. The Tracking class handles live location updates.

The relationships between these classes ensure proper communication between different modules.

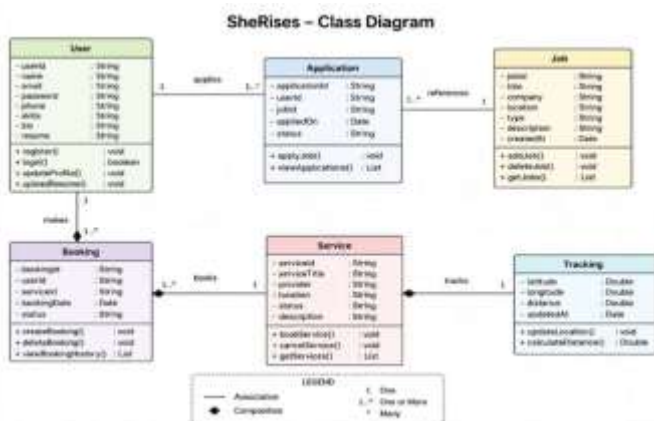


Fig. 4. Class Diagram of SHERISES

10.3 Sequence Diagram

The Sequence Diagram illustrates the chronological interaction between different components while booking and tracking a service.

The process begins when the user logs into the application. After successful authentication, the user selects a service and sends a booking request. The frontend forwards the request to the backend API, which validates the request and stores booking information in MongoDB.

Once the booking is confirmed, Socket.IO establishes a real-time communication channel between the service provider and the user. As the provider's location changes, updated coordinates are transmitted continuously to the frontend. React Leaflet displays these updates on OpenStreetMap, allowing users to monitor the provider's movement in real time.

The sequence diagram clearly explains the interaction among the frontend, backend, database, and tracking service.

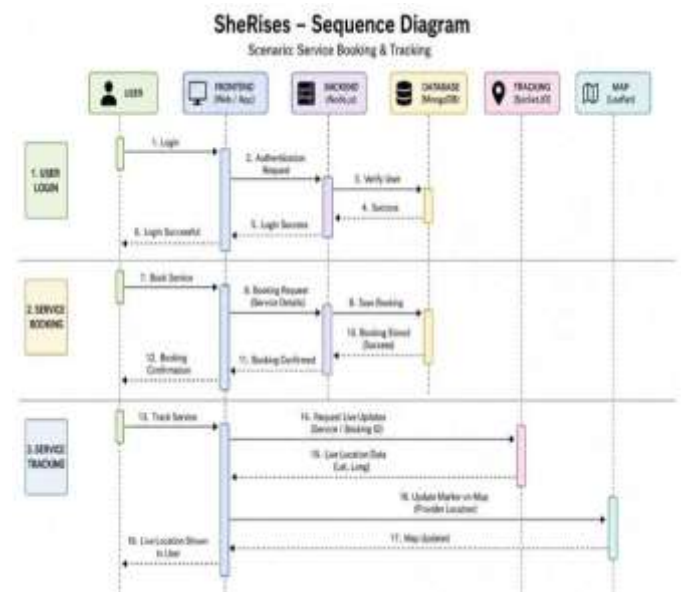


Fig. 5. Sequence Diagram of SHERISES

10.4 Deployment Diagram

The Deployment Diagram demonstrates how software components are deployed across physical hardware and network environments.

The client side consists of a web browser that runs the React.js application. The browser communicates with the Node.js and Express.js backend through HTTP requests and WebSocket connections. MongoDB stores application data, while Socket.IO enables live communication between users and service providers. React Leaflet communicates with OpenStreetMap to display geographical information.

This deployment architecture ensures modularity, scalability, and efficient communication among different components.

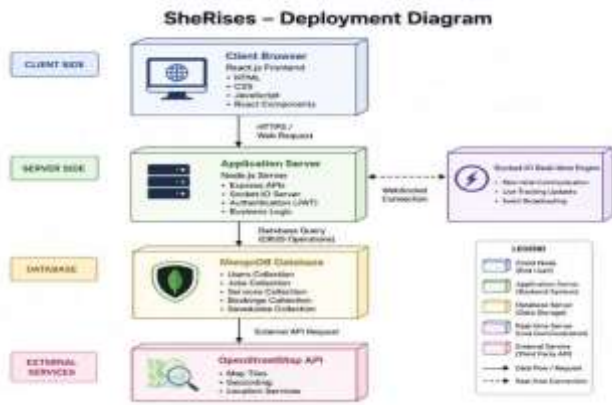


Fig. 6. Deployment Diagram of SHERISES

11. DATABASE DESIGN

A well-designed database is essential for storing and retrieving application data efficiently. SHERISES uses MongoDB, a NoSQL document-oriented database that stores information in BSON format. MongoDB was selected because of its flexibility, scalability, and ability to handle dynamic data structures.

The database consists of multiple collections that represent different modules of the application.

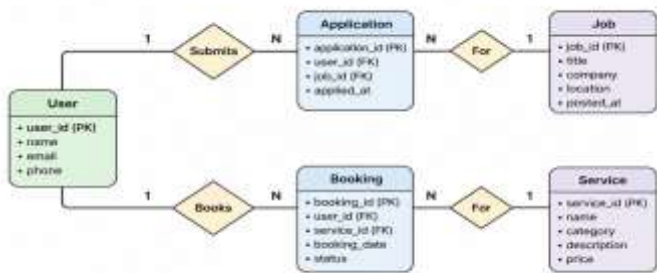


Fig. 7. Entity Relationship Diagram

11.1 User Collection

The User collection stores personal details, authentication credentials, uploaded resumes, and profile information.

Table -6: User Collection

Field	Data Type	Description
userId	ObjectId	Unique user identifier
name	String	User name

email	String	Email address
password	String	Encrypted password
phone	String	Contact number
resume	String	Resume file path

11.2 Job Collection

The Job collection stores information about available employment opportunities.

Table -7: Job Collection

Field	Data Type	Description
jobId	ObjectId	Unique job identifier
title	String	Job title
company	String	Company name
location	String	Job location
salary	String	Salary information
description	String	Job description

11.3 Service Collection

Table -8: Service Collection

Field	Data Type	Description
serviceId	ObjectId	Service ID
serviceName	String	Name of service

provider	String	Service provider
location	String	Service location
status	String	Booking status

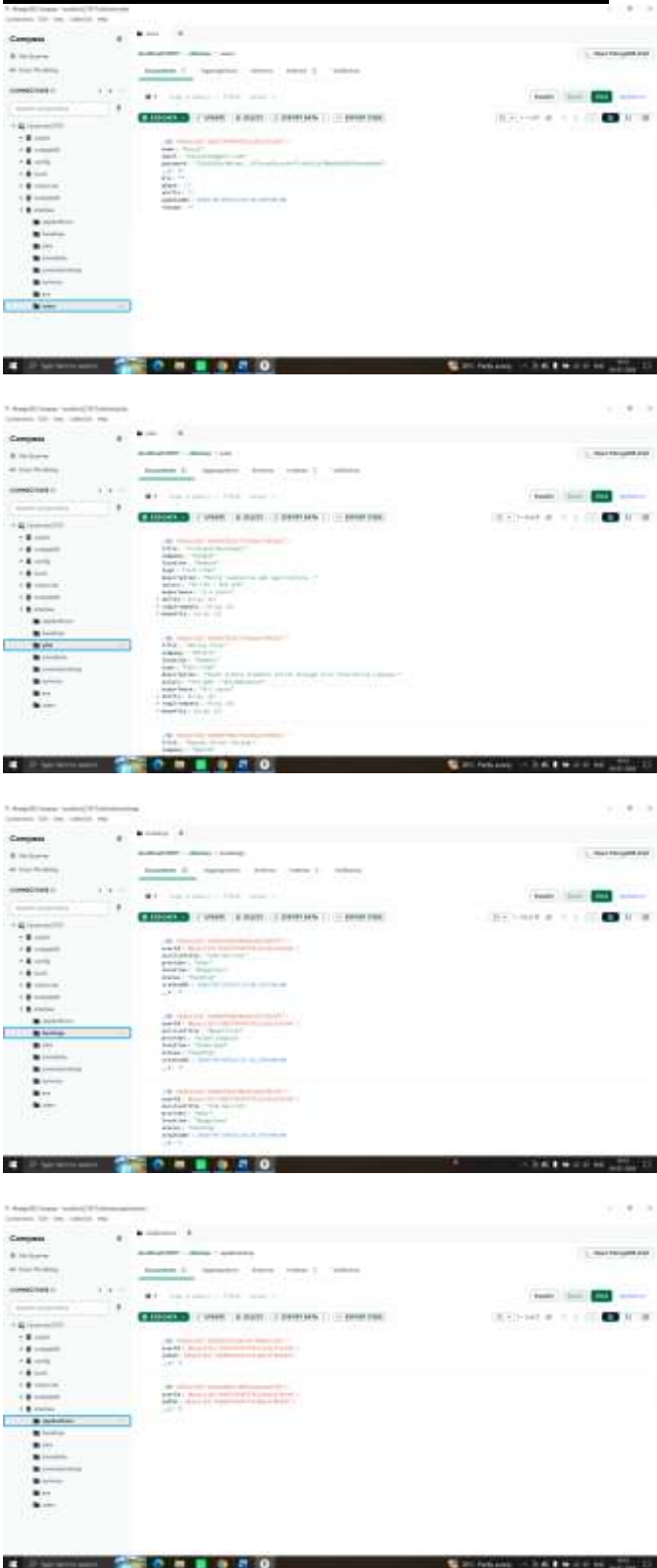


Fig. 8. MongoDB Collections Used in SHERISES

12. TOOLS AND TECHNOLOGIES USED

The successful implementation of the SHERISES platform required the integration of several modern web technologies. Each technology was selected based on its reliability, scalability, ease of development, and compatibility with real-time web applications. The combination of these technologies enabled the development of a secure, responsive, and efficient platform that satisfies the functional requirements of the system.

The frontend of the application was developed using React.js, which provides reusable components and efficient rendering through the Virtual DOM. The backend was built using Node.js and Express.js, enabling the creation of RESTful APIs for communication between the client and the server. MongoDB was selected as the database because of its flexibility in handling JSON-based documents and its seamless integration with JavaScript technologies.

Real-time tracking functionality was implemented using Socket.IO, while React Leaflet and OpenStreetMap were used to display interactive maps and live location updates. Security was enhanced through JWT authentication and bcrypt.js for password encryption.

Table -9: Technology Stack

Technology	Version/Type	Purpose
React.js	Frontend Framework	Builds responsive user interface
HTML5	Markup Language	Page structure
CSS3	Styling	Responsive layout and design
JavaScript (ES6+)	Programming Language	Frontend logic
Node.js	Runtime Environment	Backend execution
Express.js	Backend Framework	REST API development
MongoDB	NoSQL Database	Stores application data
Mongoose	ODM Library	MongoDB data modeling
Socket.IO	Real-Time Library	Live communication and tracking
React Leaflet	Mapping Library	Interactive maps
OpenStreetMap	Map Provider	Live map visualization

JWT	Authentication	Secure login sessions
bcrypt.js	Encryption Library	Password hashing
GitHub	Version Control	Source code management
Postman	API Testing Tool	Testing backend APIs
Visual Studio Code	IDE	Application development

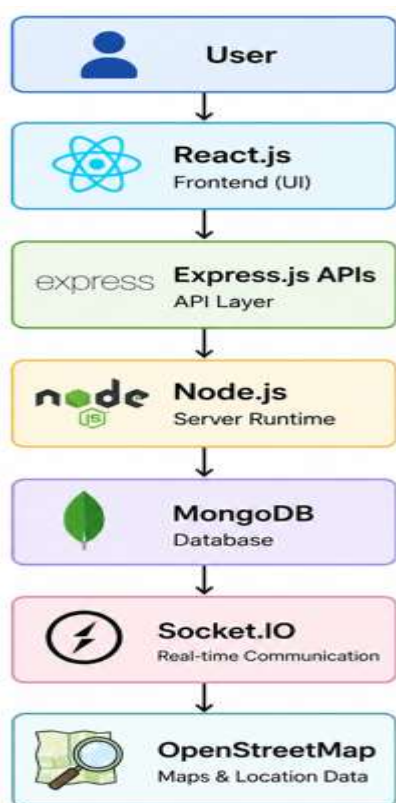


Fig. 9. Technology Stack of SHERISES

13. IMPLEMENTATION

The implementation phase involved converting the system design into a fully functional web application. The project followed a modular development approach in which each module was implemented independently before integrating it with the complete system. This approach improved code maintainability, simplified debugging, and allowed testing of individual components before final deployment.

The frontend was developed using reusable React components, while backend services were created using Express.js REST APIs. MongoDB was connected through Mongoose for efficient data storage and retrieval. Socket.IO established a persistent communication channel between users and service providers to support live tracking functionality.

The implementation is divided into the following modules.

13.1 User Registration Module

The registration module is the entry point for new users. It enables users to create an account by providing personal information such as name, email address, mobile number, and password. Before storing user information in the database, passwords are encrypted using bcrypt.js to improve security.

The system validates user inputs to ensure that duplicate email addresses are not registered and that all mandatory fields are completed correctly. After successful registration, users are redirected to the login page.

This module improves data security while providing a simple registration process.



Fig. 10. User Registration Interface

13.2 Login Module

The login module authenticates users by verifying their email and password. Once authentication is successful, a JWT token is generated and stored securely to maintain the user's session.

Protected routes prevent unauthorized users from accessing confidential pages such as profile management, service booking, and job applications. Invalid login attempts generate appropriate error messages, improving system security and usability.

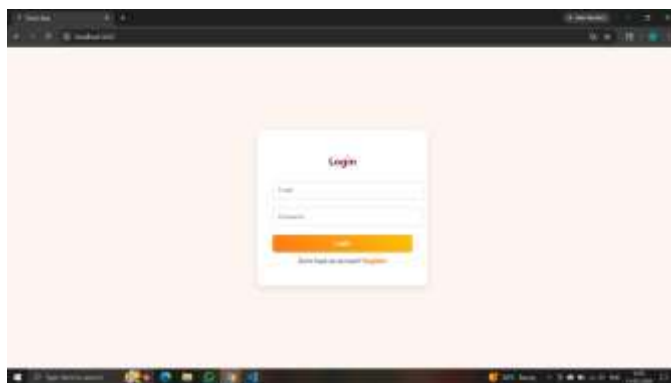


Fig. 11. User Login Interface

13.3 Dashboard Module

After successful authentication, users are redirected to the dashboard. The dashboard acts as the central navigation page from which users can access jobs, services, profile management, booking history, and live tracking.

The dashboard is designed with a responsive layout that adapts to different screen sizes, ensuring a consistent experience across desktops, tablets, and mobile devices.



Fig. 12. User Dashboard

13.4 Job Management Module

The Job Management Module enables users to search for available employment opportunities. Job details include company name, location, required skills, salary, and job description.

Users can save preferred jobs for future reference or directly apply by uploading their resumes. All job applications are stored in MongoDB and are associated with the corresponding user profile.

This module simplifies the employment process by integrating resume management and job applications within the same platform.

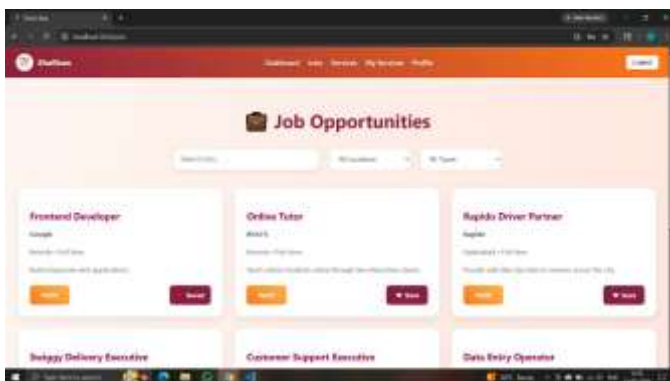


Fig. 13. Available Job Listings

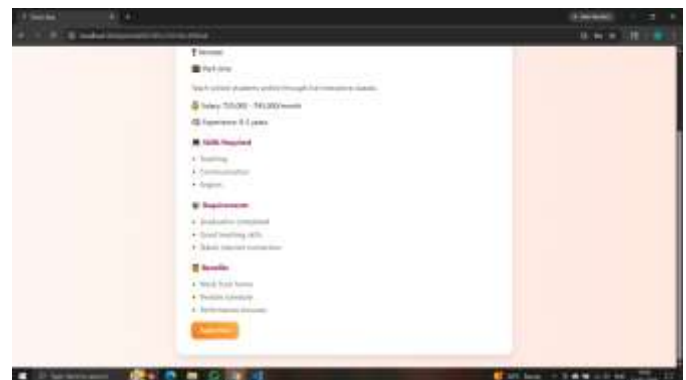
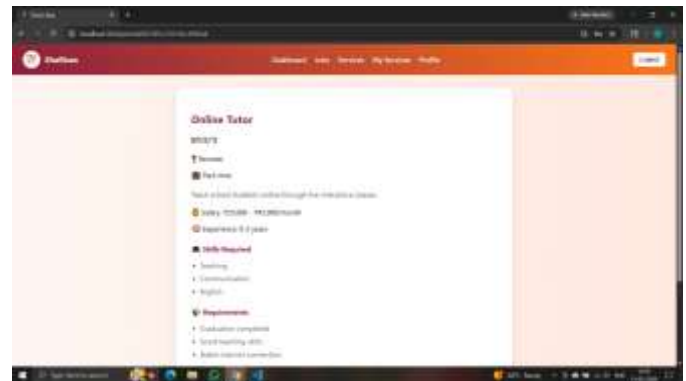


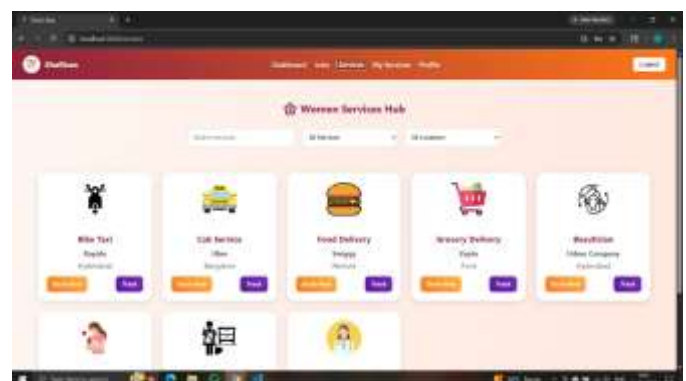
Fig. 14. Job Details and Application Interface

13.5 Service Booking Module

The Service Booking Module offers women-centric services including bike taxis, cab services, beautician services, grocery delivery, babysitting, healthcare consultation, and food delivery.

Users can conveniently browse available services, select preferred providers, and confirm bookings. Booking information is stored in MongoDB, allowing users to view their booking history at any time.

The integration of multiple services within a single platform significantly improves convenience and reduces dependency on separate applications.



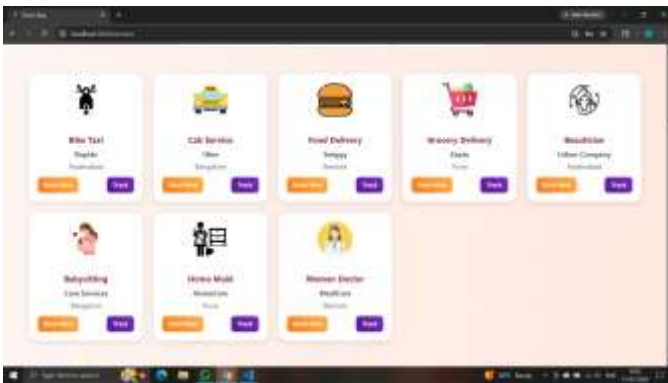


Fig. 15. Women-Centric Services

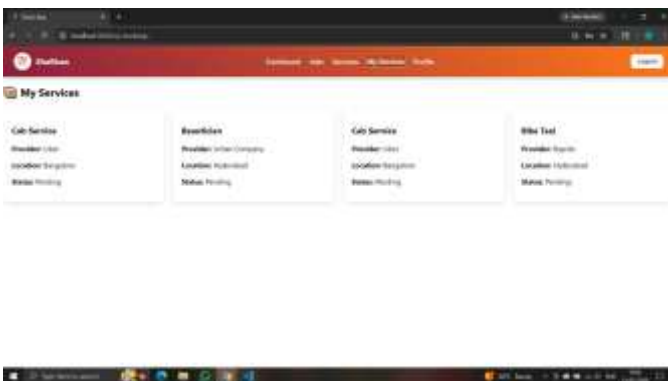


Fig. 16. Booked Services

13.6 Live Tracking Module

One of the major features of SHERISES is its real-time tracking system. After confirming a booking, users can monitor the service provider's location through an interactive map.

Socket.IO continuously exchanges location information between the backend server and connected clients. React Leaflet displays these updates using OpenStreetMap, allowing users to monitor provider movement without refreshing the page.

This feature improves transparency and user confidence while using transportation and home services.

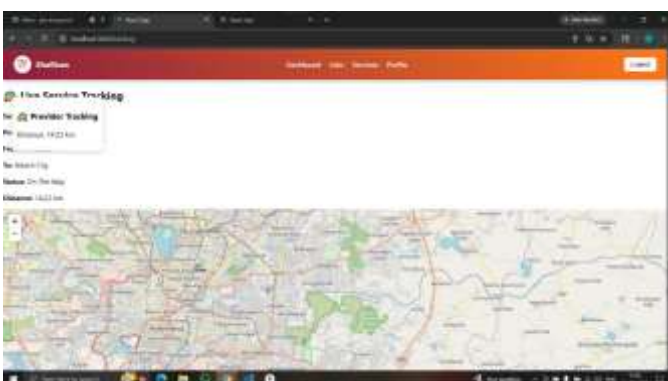


Fig. 17. Live Tracking Interface

13.7 Profile Management Module

The Profile Management Module enables users to maintain personal information and employment-related documents.

Users can update profile details, easily upload resumes, manage saved jobs, and view previous bookings. This centralized profile eliminates repeated data entry across different modules.

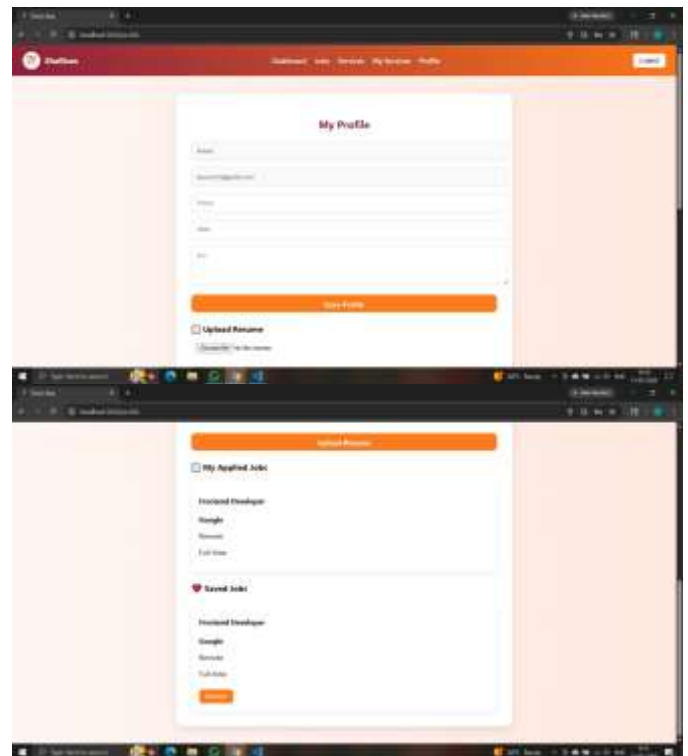


Fig. 18. User Profile Management

14. TESTING AND VALIDATION

Testing is a critical phase in software development that ensures the developed system performs as expected under different operating conditions. The SHERISES platform was evaluated through Unit Testing, Integration Testing, Functional Testing, and Performance Testing to verify the correctness, reliability, security, and responsiveness of all modules.

Each functional module—including user authentication, job management, service booking, live tracking, and profile management—was tested individually before being integrated into the complete application. The testing process confirmed that all modules interacted correctly through REST APIs and Socket.IO without functional errors.

The application was also tested on multiple web browsers to verify compatibility and responsiveness. Experimental observations indicated stable performance, secure user authentication, and efficient communication between the frontend and backend components.

14.1 Unit Testing

Unit testing was performed to validate each module independently. Individual functions such as user registration, login authentication, resume upload, service booking, and profile updates were tested to ensure correct functionality before system integration.

14.2 Integration Testing

Integration testing verified communication between React.js, Node.js, Express.js, MongoDB, and Socket.IO. Data flow between the frontend, backend, and database was successfully validated, ensuring seamless interaction among all application components.

14.3 Functional Testing

Functional testing confirmed that all user requirements were correctly implemented. Every major feature was tested using different input scenarios to ensure expected outputs and proper error handling.

Table -10: Functional Test Cases

Test Case	Expected Result	Status
User Registration	Account created successfully	Passed
User Login	User authenticated successfully	Passed
Resume Upload	Resume uploaded and stored	Passed
Job Search	Available jobs displayed	Passed
Job Application	Application submitted successfully	Passed
Service Booking	Booking confirmed successfully	Passed
Live Tracking	Real-time location updated	Passed
Profile Update	User details updated successfully	Passed
Booking History	Booking records displayed	Passed
Logout	Session terminated securely	Passed

14.4 Performance Testing

Performance testing was conducted to evaluate the responsiveness and efficiency of the application during normal usage. The experimental results indicate that the platform responds quickly to user requests while maintaining stable communication between the frontend and backend.

Table -11: Performance Metrics

Performance Metric	Measured Value	Observation
User Registration Response Time	1.4 s	Acceptable
Login Response Time	1.2 s	Fast Authentication
Average REST API Response Time	350 ms	Efficient Communication
Resume Upload Success Rate	100%	Successful Uploads
Job Application Success Rate	100%	Reliable Submission
Service Booking Success Rate	100%	Successful Booking
Live Tracking Update Interval	1–2 s	Near Real-Time Updates
Database Query Response Time	280 ms	Efficient Retrieval
Browser Compatibility	Chrome, Microsoft Edge, Firefox	Fully Supported
Responsive User Interface	Desktop & Mobile Browsers	Verified

14.5 Security Testing

Security testing was performed to verify that only authorized users could access protected resources. The following mechanisms were successfully validated:

- JWT-based authentication for secure user sessions.
- Password encryption using bcrypt.js before database storage.
- Protected API routes to prevent unauthorized access.
- Input validation to reduce invalid or malicious requests.
- Secure session management with proper logout functionality.

The successful implementation of these mechanisms enhances the confidentiality and integrity of user information stored within the platform.

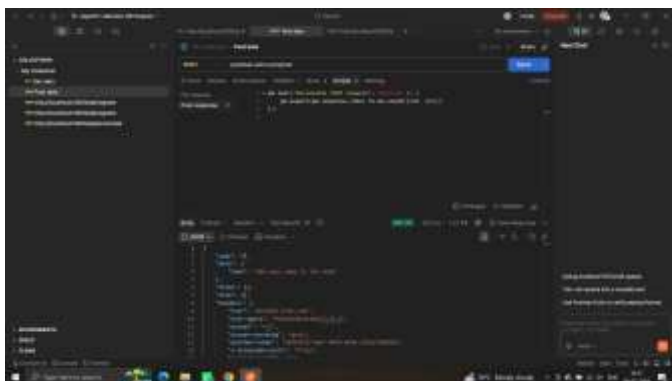


Fig. 19. Backend API Testing Using Postman

Discussion of Testing Results

The testing results demonstrate that the SHERISES platform satisfies all major functional and non-functional requirements. Authentication, job management, service booking, live tracking, and profile management operated reliably throughout the testing process. The average API response time remained below one second, and live tracking updates were reflected on the user interface within one to two seconds, providing a smooth real-time experience.

The application also maintained consistent performance across major web browsers, confirming good compatibility and responsiveness. Secure authentication, encrypted password storage, and protected API endpoints ensured that user data remained protected throughout all operations. Overall, the testing phase confirmed that SHERISES is a stable, secure, and user-friendly web application suitable for practical deployment.

15. RESULTS AND DISCUSSION

The SHERISES platform was successfully implemented as a centralized web-based application that integrates job searching, women-centric service booking, profile management, and real-time service tracking into a single system. All major modules, including user authentication, resume management, job applications, service booking, live tracking, and profile management, were tested individually and as an integrated application. The test results confirmed that each module functioned correctly and communicated efficiently through REST APIs and Socket.IO.

Unlike traditional platforms where users rely on separate applications for employment, transportation, and household services, SHERISES provides a unified environment that significantly reduces application switching. Users can search and apply for jobs, upload resumes, book trusted services, and monitor service providers without leaving the platform. This integrated workflow improves productivity and offers a more seamless user experience.

The service booking module demonstrated quick response times, allowing users to browse available services and confirm bookings within a short duration. The implementation of Socket.IO enabled real-time communication between the server and users, ensuring that location updates were reflected instantly on the tracking interface. This live tracking capability increases transparency by allowing users to monitor the movement of service providers throughout the booking process, thereby enhancing trust and safety.

The centralized profile management module further improves usability by storing user information, uploaded resumes, saved jobs, booking history, and application details in one location. This eliminates repetitive data entry across different services and simplifies profile maintenance. Secure authentication using JWT and encrypted password storage through bcrypt.js ensured that only authorized users could access protected resources, thereby improving the overall security of the application.

Overall, the experimental results indicate that SHERISES provides a practical, reliable, and scalable solution for integrating multiple women-centric services into a single web application. The modular architecture and use of modern web technologies also make the platform suitable for future enhancements such as AI-based recommendations, digital payments, emergency SOS support, and mobile application development.

Table -12: Comparative Analysis of Existing Systems and SHERISES

Feature	Existing Systems	SHERISES
Number of Applications Required	Multiple applications	Single integrated platform
Job Search	Available in separate job portals	Integrated with service platform
Service Booking	Available in separate applications	Integrated within the same system
Live Tracking	Available only in selected transport apps	Available for all booked services
Resume Management	Available only in job portals	Integrated with user profile

Profile Management	Separate profiles for different platforms	Single centralized profile
Women-Centric Services	Limited	Dedicated women-focused services
Authentication	Platform dependent	Secure JWT-based authentication
User Convenience	Moderate	High
Overall Accessibility	Fragmented	Centralized and user-friendly

Table -13: Performance Evaluation of SHERISES

Performance Metric	Observation
User Registration	Successfully completed without errors
Login Authentication	Secure JWT authentication implemented
Resume Upload	Successful file upload and storage
Job Search	Fast retrieval of job listings
Job Application	Successfully submitted and stored
Service Booking	Booking completed successfully
Live Tracking	Real-time location updates displayed
Profile Update	User information updated correctly
Booking History	Retrieved successfully
Overall System Performance	Stable and responsive

Discussion of Key Improvements

The implementation of SHERISES resulted in several practical improvements over conventional standalone applications:

- **Reduced Application Switching:** Users no longer need separate applications for job searching, transportation, and women-centric services, reducing complexity and saving time.
- **Faster Service Booking:** A unified interface and optimized backend APIs enable quicker booking and confirmation of services.
- **Enhanced Transparency:** Real-time location tracking through Socket.IO and OpenStreetMap provides continuous updates, increasing user confidence during service delivery.
- **Improved User Convenience:** A single dashboard allows users to access jobs, services, tracking, and profile management without navigating multiple platforms.
- **Centralized Profile Management:** Personal details, resumes, saved jobs, applications, and booking history are maintained in one secure profile, eliminating redundant data entry.
- **Better Security:** JWT-based authentication and bcrypt password hashing ensure secure access to the platform and protect sensitive user information.

16. FUTURE SCOPE

Although the SHERISES platform successfully achieves its objectives, several enhancements can be incorporated in future versions to improve functionality and user experience.

Potential future improvements include:

- AI-based job recommendation system using machine learning.
- Online payment gateway integration for service bookings.
- SOS emergency alert feature for enhanced safety.
- Mobile applications for Android and iOS.
- Push notifications for bookings and job updates.
- Real-time chat between users and service providers.
- Voice-enabled navigation for improved accessibility.
- Multi-language support to reach a wider user base.
- Cloud deployment for improved scalability and availability.
- Analytics dashboard for administrators to monitor platform usage and service trends.

These enhancements would make the platform more intelligent, accessible, and suitable for large-scale deployment.

17. CONCLUSION

The SHERISES platform was developed as an integrated web application to support women by combining employment opportunities, women-centric services, profile management, and real-time service tracking within a single system. Unlike conventional applications that focus on individual functionalities, SHERISES provides a unified solution that

simplifies daily activities while emphasizing safety, accessibility, and convenience.

The application showcases the practical use of modern web technologies such as React.js, Node.js, Express.js, MongoDB, Socket.IO, and React Leaflet. Secure authentication using JWT and password encryption through bcrypt.js further enhance the reliability of the platform.

The evaluation results confirmed that all modules function correctly and communicate effectively, resulting in a responsive and user-friendly system. The modular architecture also provides flexibility for future enhancements, making SHERISES a scalable solution that can evolve with emerging technologies.

Overall, the project effectively achieves its objective of creating a centralized digital platform that empowers women through secure employment opportunities, trusted services, and transparent real-time tracking.

Unlike conventional standalone applications, SHERISES demonstrates the practical implementation of a unified women empowerment platform that combines employment, trusted services, secure authentication, and real-time tracking within a single scalable web application.

ACKNOWLEDGEMENT

The authors express their sincere gratitude to the Department of Computer Science and Engineering, JNTUH University College of Engineering, for providing the opportunity and necessary facilities to carry out this project successfully. We are deeply thankful to our project guide and faculty members for their valuable guidance, constructive suggestions, and continuous encouragement throughout the development of this work. We also extend our appreciation to our classmates and friends for their support during the implementation and testing phases of the project. Finally, we express our heartfelt gratitude to our parents and family members for their constant motivation, encouragement, and unwavering support, which played a significant role in the successful completion of this project.

REFERENCES

[1] Alex Banks and Eve Porcello,
Learning React: Modern Patterns for Developing React Apps,
2nd Edition,
O'Reilly Media,
2020.

[2] Ethan Brown,
Web Development with Node and Express: Leveraging the JavaScript Stack,
2nd Edition,
O'Reilly Media,
2019.

[3] Jon Duckett,
HTML and CSS: Design and Build Websites,
John Wiley & Sons,
2011.

[4] Jon Duckett,
JavaScript and jQuery: Interactive Front-End Web Development,
John Wiley & Sons,
2014.

[5] Ramez Elmasri and Shamkant B. Navathe,
Fundamentals of Database Systems,
7th Edition,
Pearson Education,
2017.

[6] Kristina Chodorow,
MongoDB: The Definitive Guide,
3rd Edition,
O'Reilly Media,
2019.

[7] A. K. Sood and R. J. Enbody,
"Targeted Cyberattacks: A Superset of Advanced Persistent Threats,"
IEEE Security & Privacy,
Vol. 11, No. 1, pp. 54–61,
2013.

[8] J. Brooke,
"SUS: A Quick and Dirty Usability Scale,"
Usability Evaluation in Industry,
Taylor & Francis,
pp. 189–194,
1996.

[9] M. Satyanarayanan,
"Pervasive Computing: Vision and Challenges,"
IEEE Personal Communications,
Vol. 8, No. 4, pp. 10–17,
2001.

[10] H. Gupta and P. Sharma,
"GPS-Based Women Safety System Using IoT,"
International Journal of Innovative Technology and Exploring Engineering (IJITEE),
Vol. 8, No. 6,
2019.

[11] S. S. Chavan and P. R. Deshmukh,
"Women Safety System Using GPS and GSM,"
International Journal of Advanced Research in Computer Engineering & Technology,

Vol. 5, No. 3,
2016.

[12] N. K. Gondaliya and V. R. Rathod,
"Design and Implementation of Women Safety
Application Using Android,"
*International Journal of Scientific Research in
Computer Science, Engineering and Information
Technology*,
Vol. 5, No. 2,
2019.

[13] M. Zaharia et al.,
"Apache Spark: A Unified Engine for Big Data
Processing,"
Communications of the ACM,
Vol. 59, No. 11, pp. 56–65,
2016.

[14] S. Newman,
Building Microservices,
2nd Edition,
O'Reilly Media,
2021.

[15] React Team,
React Documentation,
Available: <https://react.dev>

[16] Node.js Foundation,
Node.js Documentation,
Available: <https://nodejs.org/docs>

[17] Express.js Team,
Express.js Documentation,
Available: <https://expressjs.com>

[18] MongoDB Inc.,
MongoDB Documentation,
Available: <https://www.mongodb.com/docs>

[19] Socket.IO Team,
Socket.IO Documentation,
Available: <https://socket.io/docs>

[20] React Leaflet Contributors,
React Leaflet Documentation,
Available: <https://react-leaflet.js.org>

[21] OpenStreetMap Foundation,
OpenStreetMap Documentation,
Available: <https://www.openstreetmap.org>

[22] Auth0 Inc.,
JSON Web Token (JWT) Documentation,
Available: <https://jwt.io>