Winter Internship – Task 2 For Team (Kappa)

Smart Service Booking & Management System(SSBMS)

Project Documentation – MERN Stack

1. Project Duration

17th Jan to 16 Feb , 2026

2. Project Overview

SSBMS is a real-world MERN stack web application designed to allow customers to book services online while enabling service providers and administrators to manage operations efficiently.

3. User Roles

Admin: Manage users, approve providers, monitor system.
Service Provider: Manage services and bookings.
Customer: Browse services and create bookings.

4. Functional Requirements

• Secure authentication using JWT
• Service CRUD operations
• Booking creation and status management
• Admin approval workflows
• Responsive dashboards

5. Non-Functional Requirements

• Secure APIs
• Clean folder structure
• Scalable architecture
• Error handling and validation

6. Technology Stack

Frontend: React, Redux Toolkit, Tailwind CSS
Backend: Node.js, Express.js
Database: MongoDB (Mongoose)
Deployment: Vercel, Render, MongoDB Atlas

7. Database Schemas

User: name, email, password, role, isApproved
Service: title, description, price, duration, providerId
Booking: customerId, serviceId, providerId, date, status

8. API Modules

Authentication APIs, Service APIs, Booking APIs, Admin APIs following REST standards.

9. Folder Structure

Frontend: components, pages, redux, services, hooks
Backend: controllers, models, routes, middleware, utils

10. Development Timeline (4 Weeks)

Week 1: Setup & Authentication
Week 2: Core Features & APIs
Week 3: Dashboards & UI
Week 4: Testing, Deployment & Documentation

11. Security Guidelines

• Hash passwords using bcrypt
• Protect routes using JWT
• Use environment variables

12. Testing & Deployment

Test APIs using Postman, validate frontend flows, deploy backend and frontend to cloud platforms.

13. Intern Deliverables

• Working web application
• GitHub repository
• Live deployed link
• Final presentation

Submission Requirements

Submit your complete project via Google Classroom. Upload all necessary code, files, and folders required to run your project properly. Incomplete or missing files may result in rejection.

Evaluation Criteria

Structure and completeness, responsiveness, code cleanliness, UI clarity, and Git usage.

Important Notes

Plagiarized or template-based work will be rejected. Fake projects or experience will negatively impact evaluation.Focus on clarity and quality over visual effects.

This task evaluates fundamentals, discipline, and attention to detail. Build something simple, honest, and clean.