Synopsis

On

FurniMart – B2B E-commerce Furniture Platform

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE ENGINEERING DEPARTMENT

Submitted To:

SIGNATURE

Mr.Neeraj Raheja

Project Coordinator

Submitted By:

Jagjeet Kumar	**11222820 (E1)**
Sarvesh Kumar	**11222825 (E2)**

FurniMart – B2B E-commerce Furniture Platform

CHAPTER 1: INTRODUCTION

1.1 Project Overview:

FurniMart is a B2B (Business-to-Business) e-commerce platform specifically designed for the furniture industry. The primary aim of this project is to digitize and simplify the process of bulk furniture procurement, customization, and delivery. The system connects **manufacturers** and **wholesalers** on a centralized platform, monitored and managed by an **admin** who verifies manufacturer authenticity and maintains platform integrity.

This system enables **state-to-state delivery** of furniture items with a future vision of scaling it to **international logistics**. The manufacturers can manage their product listings, customize orders based on client requirements, and receive bulk orders from verified wholesalers. The wholesalers, in turn, can communicate directly with manufacturers using **WebRTC** for pre-order discussions and place orders with a **minimum quantity of 20 items**. The payment is processed in advance through an integrated **payment gateway**.

FurniMart stands out for its smart integrations, premium service offerings, and future-ready design, offering innovative features such as AI-powered business recommendations, global logistics support, and enterprise solutions.

1.2 Background:

Traditional bulk procurement in the furniture industry lacks digital support and transparency. Most interactions are still manual, time-consuming, and rely on third-party agents or offline methods. There is no easy way to verify a manufacturer or request product customization at scale.

Platforms like Alibaba offer global B2B services, but lack niche-specific features tailored to the furniture sector in India. FurniMart fills this gap by offering **verified onboarding, real-time communication, customizable orders, and structured delivery pipelines** specifically focused on furniture commerce.

1.3 Problem Statement

- No transparent platform for verified furniture manufacturers and buyers.
- Lack of real-time communication between manufacturers and wholesalers.
- No provision for easy product customization and advance order tracking.
- Traditional B2B methods are slow, scattered, and unscalable.
- Absence of specialized admin controls and bulk order thresholds.

1.4 Proposed Solution

FurniMart aims to address these problems with a complete B2B furniture ecosystem using MERN stack and DevOps automation. The key features include:

- **Role-Based Access:** Admin, Manufacturer, and Wholesaler roles with custom dashboards.
- **Admin Verification System:** Only admin can verify genuine manufacturers.
- **WebRTC Communication:** Enables wholesalers to interact with manufacturers before placing orders.
- **Advance Payment Gateway:** Ensures pre-payment before production.
- **Minimum Order Quantity Rule:** Only bulk orders (20+ items) allowed.
- **AI-Powered Recommendations:** Premium users get insights and tools to expand business.
- **Premium Subscriptions:** Access to enterprise features like global logistics, branding, and analytics.
- **Future Scope:** Expand from state-level delivery to global cross-border furniture supply

CHAPTER 2:OBJECTIVES

2.1 Primary Objectives

1. **Role-Based Access System**
   Implement distinct login systems for Manufacturer, and Wholesaler. Each role will have a separate dashboard and set of permissions, ensuring proper segregation of duties and security.
2. **Admin-Verified Onboarding**
   Only verified manufacturers will be allowed to sell on the platform. Admins will review submitted details and documents before approving any manufacturer to prevent scams or unauthentic entries.
3. **Manufacturer Dashboard for Product Management**
   A dedicated dashboard for manufacturers where they can upload new products, set descriptions, add images, track order metrics, and manage inventory in real-time.
4. **Wholesaler Dashboard for Order Placement**
   Wholesalers can browse available furniture, communicate with manufacturers, and place bulk orders (minimum 20 items), while also managing their cart and purchase history.
5. **WebRTC-Based Chat Communication**
   Allow real-time communication via Chat between wholesaler and manufacturer before finalizing orders, helping in better customization discussion and trust-building.
6. **Payment Gateway with Advance Payment Rule**
   A secure, integrated payment system that mandates advance payments from wholesalers to avoid order abandonment. No cash-on-delivery — only verified online payments.
7. **Minimum Order Quantity Validation**
   A built-in validation logic that prevents wholesalers from placing orders less than 20 units, enforcing the B2B nature of the platform.
8. **Real-Time Order Tracking System**
   Both manufacturer and wholesaler can monitor order status — from customization → dispatch → delivery — with milestone-based updates.

2.2 Secondary Objectives

1. **Premium Subscription Modules**
   Manufacturers will have an option to unlock premium features (like analytics, logistics expansion, hiring tools) by subscribing to paid plans — boosting revenue and scalability.
2. **Staff Hiring System**
   Manufacturers can post job roles and manage hiring through their dashboard — future module for expanding their team with skilled professionals.
3. **Admin Analytics Dashboard**
   Admins will have access to visual reports on platform activity — number of orders, active users, premium subscribers, and verification pending requests.
4. **Scalable State-to-Country Delivery Planning**
   The backend will be designed in a way to easily support international expansion in future — with modular delivery zones and logistic partner integration.
5. **Secure Session Management**
   Implement encrypted session handling using JWT tokens and refresh tokens, ensuring user authentication and prevention of unauthorized access.
6. **Notification and Alert System**
   Push or in-app notifications for order updates, manufacturer messages, admin approvals, and payment confirmation will enhance engagement.
7. **Version Control Collaboration**
   The project will be built using GitHub collaboration between team members (Jagjeet & Sarvesh), ensuring clean development, pull requests, and CI/CD pipelines.

CHAPTER 3:HARDWARE AND SOFTWARE REQUIREMENTS

3.1 Hardware Requirements

- **Client-Side Devices**
  Any device capable of running a modern browser can use the platform (Laptop/Desktop/Mobile/Tablet).
- **Development System Requirements**
  - Minimum RAM: 8 GB
  - Processor: Intel i5 or equivalent
  - Disk Space: Minimum 10 GB for local development and Docker container support
  - Stable internet connection (for GitHub sync and deployment)
- **Production Server Requirements**
  - 16 GB RAM for handling multiple microservices
  - 4-core CPU
  - SSD-based cloud storage for fast DB access
  - Cloud Hosting with HTTPS/SSL (AWS, Vercel, or Render)

3.2 Software Requirements

- **Operating System**
  Cross-platform (Windows, macOS, Linux) for development
- **Browser Compatibility**
  Chrome, Firefox, Safari, Edge (latest versions)
- **Third-party Services**
  - Cloudinary or AWS S3 for product image uploads
  - Razorpay or Stripe for secure payment integration
  - WebRTC APIs for real-time video
  - GitHub for source control and CI/CD
  - Postman for API testing

3.3 Technology Stack (Full Description)

Layer	Technology Used	Description
Frontend	React.js + TailwindCSS	Fast, component-based UI + modern styling
Backend	Node.js + Express	REST API development with clean routing and middleware support
Database	MongoDB	NoSQL for flexibility
Authentication	JWT	Stateless, secure session-based login with role checks
Communication	WebRTC	Peer-to-peer real-time chat connection between wholesaler and seller
DevOps	GitHub Actions	Automated build, test, and deploy pipelines
Hosting	AWS / Vercel	Scalable cloud platform for frontend & backend deployment

3.4 Development Tools

- **Code Editor:** Visual Studio Code
- **Version Control:** Git and GitHub
- **API Testing Tool:** Postman
- **Database GUI:** MongoDB Compass
- **Browser Tools:** Chrome DevTools
- **Deployment Tools:** Docker (optional), GitHub Actions, AWS EC2/S3

CHAPTER 4:METHODOLOGY TO BE USED

4.1 Project Methodology

The project is developed using the **Agile Software Development Lifecycle (SDLC)**. Agile promotes an iterative and incremental approach, where continuous feedback, testing, and collaboration help us stay flexible and efficient.

This allows the team (Jagjeet & Sarvesh) to plan, design, develop, test, and deploy in short cycles (sprints), track progress using GitHub issues, and make continuous improvements.

4.2 Project Phases

1. **Requirement Gathering & Planning**
   - Define user roles (Admin, Manufacturer, Wholesaler)
   - Feature list preparation
   - Flow diagram and use-case identification
2. **UI/UX Design**
   - Design of user dashboards and product cards
   - Create wireframes using Figma or simple sketches
   - Choose appropriate color schemes and layouts
3. **Frontend Development**
   - React.js with Tailwind CSS for responsive components
   - State management using Redux (if needed)
   - Dynamic category rendering & order forms
4. **Backend Development**
   - Node.js + Express for REST APIs
   - MongoDB for storing user/product/order data
   - JWT for secure authentication
   - Role-based access protection
5. **Integration**
   - WebRTC API integration for real-time video calls
   - Razorpay or Stripe for payment gateway
   - File uploads (Cloudinary / S3)
6. **Testing & Debugging**
   - Unit testing with Postman
   - UI testing in browser
   - Real-world scenario simulation with test accounts
7. **Deployment**
   - Hosting on AWS or Render
   - GitHub Actions for automatic CI/CD

4.3 Development Tools

- **IDE:** Visual Studio Code
- **Version Control:** Git & GitHub
- **Design:** Figma / Canva (optional)
- **Browser Debugging:** Chrome DevTools
- **API Testing:** Postman
- **CI/CD:** GitHub Actions
- **Deployment:** AWS EC2 / Vercel

CHAPTER 5: FEATURES AND MODULES

5.1 Core Modules

1. **User Roles & Authentication**
   - Manufacturer, Wholesaler
   - JWT-based login system
   - Role-based dashboard routing
2. **Manufacturer Dashboard**
   - Upload product info (images, specs, material)
   - Track orders, revenue, customer interactions
   - Access analytics (if subscribed to premium)
3. **Wholesaler Dashboard**
   - Browse available products
   - Filter based on category/material
   - Minimum order validation (20 items)
   - Advance payment & real-time communication
4. **Admin Dashboard**
   - Approve or reject manufacturer requests
   - Monitor total orders, platform metrics
   - View user reports, logs, system health

5.2 Real-Time Communication

- Built using **WebRTC**
- Enables video chat between wholesaler and manufacturer
- Helps in customization and trust-building before placing large orders

5.3 Payment System

- Integrated Razorpay or Stripe API
- Wholesaler must pay in advance
- Secures manufacturer interest
- Generates transaction history and invoices

5.4 Premium Manufacturer Features

Only available with **Enterprise Plan** or **Annual Premium**, these features help manufacturers expand their business:

1. **AI Business Intelligence**
   new features or innovations are launched regularly.
2. **Hiring Dashboard**
   Post job openings, filter candidates, shortlist for interviews.
3. **Global Logistics**
   Access shipping support to 190+ countries.
4. **Brand Customization**
   Add custom logos, white-labeling support for catalogs.

CHAPTER 7: REFERENCES

- React.js Official Docs – https://reactjs.org
- Node.js Official Docs – https://nodejs.org
- MongoDB Manual – https://www.mongodb.com/docs
- WebRTC Guide – https://webrtc.org
- Razorpay Integration – https://razorpay.com/docs/
- JWT Authentication – https://jwt.io
- GitHub Actions – https://docs.github.com/actions
- Agile Methodology – https://www.agilealliance.org
- Alibaba Official Platform – https://www.alibaba.com
- Tailwind CSS – https://tailwindcss.com