Project Documentation

Store manager: Keep tracking inventory

1. Introduction

- **Project Title:** Store manager: Keep tracking inventory
- **Team ID:** NM2025TMID46577
- **Team Leader:** S.Logashree & logashreelogashree78@gmail.com
- **Team Members:**

  - P.Vidhya & vidhya7904169249@gmail.com
  - R.Miruthula & ms.mirthula.rajendran@gmail.com
  - T.Banupriya & banupriyakural@gmail.com

2. Project Overview

- **Purpose:** The purpose of this project is to provide a modern, user-friendly front-end system that helps store managers efficiently track, manage, and visualize inventory data. It simplifies stock management by offering clear insights into product availability, low stock alerts, and out-of-stock notifications, ensuring better decision-making and smoother store operations.
- **Features:**

  - Dashboard Overview – Displays total, available, low stock, and out-of-stock items at a glance.
  - Inventory Table – Manage products with images, categories, stock status, and search option.
  - Add / Edit Product – Simple modal form to add new items or update existing ones.
  - Stock Status Indicators – Color-coded badges for quick tracking of inventory levels.
  - Visual Analytics – Interactive Chart.js graph to monitor stock distribution.

3. Architecture

- **Frontend:** React.js with Bootstrap and Material UI
- **Backend:** Node.js and Express.js managing server logic and API endpoints
- **Database:** MongoDB stores user data, project information, applications, and chat messages

4. Setup Instructions

- **Prerequisites:**
  - Node.js
  - MongoDB
  - Git
  - React.js
  - Express.js – Mongoose – Visual Studio Code
- **Installation Steps:**

```
# Clone the repository git clone

# Install client dependencies cd client
npm install

# Install server dependencies cd
../server npm install
```

5. Folder Structure

```
SB-Works/
|-- client/                 # React frontend
      |__components/
        L__ pages/
|__ server/                 # Node.js backend
      |__routes/
         |__ models/
           |__ controllers/
```

6. Running the Application

- **Frontend:**

```
cd    client npm
```

start • **Backend:**

```
cd server npm
start
```

- **Access:** Visit http://localhost:3000

7. API Documentation

- **User:**
  - /api/user/register
  - /api/user/login
- **Projects:**
  - /api/projects/create
  - /api/projects/:id • **Applications:** /api/apply
- **Chats:**
  - /api/chat/send – /api/chat/:userId

8. Authentication

- JWT-based authentication for secure login
- Middleware protects private routes

9. User Interface

- Landing Page
- Freelancer Dashboard
- Admin Panel
- Project Details Page

10. Testing

- Manual testing during milestones
- Tools: Postman, Chrome Dev Tools

11. Screenshots or Demo

12. Known Issues

13. Future Enhancements