*Introduction*

* Project Title: Store Manager
* Team ID: (Fill in if needed)
* Team Leader:
* Swetha C & [swethaswetha.c2006@gmail.com](mailto:swethaswetha.c2006@gmail.com)
* Team Members:
* Monisha E.L & [elmonisha07@gmail.com](mailto:elmonisha07@gmail.com)
* Saranya K & [saranyadevikarthikeyan@gmail.com](mailto:saranyadevikarthikeyan@gmail.com)

*Project Overview*

Purpose:

* Store Inventory Manager is a web-based system that helps businesses manage their product stock efficiently. Users can add, search, and categorize products with details such as name, price, and image.
* Features:
* Product Search by category and name
* Add new product (name, price, image)
* Category-based organization (e.g., Electronics, Grocery, Clothing, etc.)
* User-friendly UI with product thumbnails

*Architecture*

* Frontend: HTML, CSS, JavaScript (Vanilla or Framework if used)
* Backend: (If applicable – Node.js / Flask / Django / PHP – specify what you used)
* Database: (If used – MySQL, MongoDB, or Local JSON storage)

*Setup Instructions*

Prerequisites:

Web Browser

VS Code / Any IDE

Node.js (if backend included)

Installation Steps:

Clone or download the project

Open project folder in VS Code

Run with Live Server (for frontend only) OR npm start (if backend included)

*Folder Structure*

Copy code

Store-Inventory-Manager/

|-- index.html

|-- style.css

|-- script.js

|-- images/

|-- server/ (if backend used)

|-- app.js

|-- routes/

|-- models/

*Running the Application*

Frontend:

Open index.html in browser (or run via Live Server).

Backend (if used):

Copy code

cd server

npm start

Access: Visit <http://127.0.0.1:5500/index.html>

*API Documentation (If backend exists)*

Products:

* POST /api/products/add → Add new product
* GET /api/products → Fetch all products
* GET /api/products/:id → Fetch product by ID

*Authentication (Optional, if you added login system)*

JWT/session-based authentication for user access.

*User Interface*

* Product Search Bar
* Add Product Form (Name, Price, Image, Category)
* Product Display Grid with Images

*Testing*

* Manual testing of adding/searching products.
* Tools: Browser Dev Tools, Postman (if backend).

*Screenshots or Demo*

(Add screenshots like the one you shared.)

*Known Issues*

* Limited UI responsiveness on mobile (if applicable).
* No database integration in frontend-only version.

*Future Enhancements*

* User login system
* Advanced filters & sorting
* Stock quantity management
* Admin dashboard

