A black background with text and a picture of a person

AI-generated content may be incorrect. A black and white logo

AI-generated content may be incorrect.

**9530**

**St. MOTHER THERESA ENGINEERING COLLEGE**

COMPUTER SCIENCE ENGINEERING

**NM-ID**: E581886B3B1AF1CAFA5414A986D0DE9B

**REG NO**: 953023104016

**DATE**: 22-09-2025

**Completed the project named as**

**Phase 3**

FRONT END TECHNOLOGY

**CHAT APPLICATION UI**

SUBMITTED BY,

S.Aswin

7538849870

**Phase 3 — MVP Implementation (Deadline – Week 8)**

**1. Project Setup**

Initialize the project environment with required frontend framework (React / Angular / Vue).

Configure package manager (npm/yarn) and install dependencies.

hbSet up development tools such as ESLint, Prettier, Babel, and Webpack (if needed).

Create basic folder structure for components, services, assets, and utilities.

Configure environment variables (API base URLs, database configs).

**2. Core Features Implementation**

User Authentication: Login, registration, and logout functionality.

Product Catalog: Display of products with images, descriptions, and prices.

Search & Filter: Ability to find products quickly using keywords, categories, or price range.

Shopping Cart: Add/remove items, update quantity, and view cart summary.

Checkout Process: Order confirmation flow (address, payment option placeholder, order review).

**3. Data Storage (Local State / Database)**

Local State Management: Use React Context API/Redux for managing UI state (cart, authentication, etc.).

**Database Integration:**

Store product data, user details, and orders in a database (e.g., MongoDB, MySQL, or Firebase).

Implement CRUD operations (Create, Read, Update, Delete) for products and orders.

APIs: Connect frontend with backend APIs for fetching/storing product and order data.

**4. Testing Core Features**

Unit testing for individual components (using Jest/Mocha).

Integration testing for critical flows like login, adding to cart, and checkout.

Manual testing of UI responsiveness (mobile, tablet, desktop).

Debugging and fixing functional issues before deployment.

**5. Version Control (GitHub)**

Initialize Git repository and link to GitHub.

Create feature branches for modular development (e.g., feature/cart, feature/auth).

Commit changes with proper messages.

Pull requests and code reviews to ensure quality and avoid conflicts.

Maintain version history for easy rollback and collaboration.