MOBILE PROGRAMMING LAB

Topic: Product Management & Shopping Cart Application

Objective

Students will build a complete mobile application with features such as user registration & login, product display and management, shopping cart, and revenue statistics. The goal is to practice data handling, UI design, navigation, and product list manipulation.

Required Features

1) User Registration

- Users enter: Full Name, Email, Password, Confirm Password.
- Validate input:
  - Email format must be correct.
  - Password must match Confirm Password.
- Save user data to local database or server.

2) User Login

- Allow login using registered Email and Password.
- Validate credentials with stored data.
- On success, navigate to the product list screen.
- Provide “Remember Password” option using SharedPreferences.

3) Product List

- Display a list of products with: Name, short description, price, and image (if available).
- Use RecyclerView (Android) to display the list.

4) Product Management (CRUD)

- **Add Product**: Enter name, description, price, image (URL or from local storage).
- **View Details**: Tap an item to view full product details.
- **Update Product**: Edit product information from detail or list view.
- **Delete Product**: Remove a product from the list.

5) Search Products

- Enable searching by product name.
- Display real-time search results while typing.

6) Sort Products

- Sort product list by price:
  - Ascending
  - Descending

7) Shopping Cart

- Allow users to add products to the cart.
- Display cart items with quantity and total price.
- Update item quantity directly in the cart.
- Remove items from cart.
- Provide a “Checkout” button to simulate order placement.

8) Revenue Statistics

- Record orders after checkout.
- Calculate and display total revenue from completed orders.
- Optionally allow filtering revenue by:
  - Day
  - Month
  - Year

UI Requirements

- User-friendly and intuitive interface.
- Use Navigation and ActionBar.
- Each feature should be placed on a separate screen.
- Show messages or alerts for success and error actions.

Technical Requirements

- Students may choose one of the following storage options:
  - SQLite (local storage)
  - Mock API (JSON Server)
  - MongoDB
  - Firebase
- Organize code into clear components: forms, lists, details, cart, revenue, etc.