Primary Requirements for the Budgeting App

Functional Requirements

User Account Management

- The system shall allow users to register using email and password.
- The system shall allow users to log in and log out securely.
- The system shall allow users to reset forgotten passwords.
- The system shall store user profile details (name, email, currency preference).

Income Management

- The system shall allow users to add income records.
- The system shall allow users to edit and delete income entries.
- The system shall store income amount, source, and date.
- The system shall display total income per selected period (daily, weekly, monthly).

Expense Management

- The system shall allow users to add expense records.
- The system shall allow users to edit and delete expense entries.
- The system shall store expense amount, category, description, and date.
- The system shall support quick-add expense entry.
- The system shall allow users to attach optional notes to expenses.

Expense Categorization

- The system shall provide predefined categories (e.g., Food, Transport, Rent, Utilities, Entertainment, Education, Health).
- The system shall allow users to create custom categories.
- The system shall allow users to edit or delete custom categories.
- The system shall assign categories manually or automatically.

Budget Management

- The system shall allow users to create budgets for each category.
- The system shall allow users to set budget periods (weekly or monthly).
- The system shall display remaining budget amounts.
- The system shall notify users when they approach or exceed budgets.

Financial Goals

- The system shall allow users to create savings goals.
- The system shall store goal name, target amount, and deadline.
- The system shall display progress toward each goal.
- The system shall allow users to modify or delete goals.

Reports and Visualization

- The system shall generate spending reports by category.
- The system shall display charts (pie chart, bar chart, line graph).
- The system shall show income vs expenses comparison.
- The system shall allow filtering by date range.

Notifications and Reminders

- The system shall send reminders to log daily expenses.
- The system shall send alerts for budget limit thresholds.
- The system shall allow users to enable or disable notifications.

Data Storage and Backup

- The system shall store data locally on the device.
- The system shall synchronize data to cloud storage (optional).
- The system shall allow users to export data as CSV or PDF.

Non-Functional Requirements

Usability

- The application shall have a simple and intuitive user interface.
- The application shall support light and dark modes.
- The application shall require no more than three taps to add an expense.

Performance

- The application shall load within 3 seconds.
- The application shall handle at least 10,000 records without lag.

Security

- The system shall encrypt user passwords.
- The system shall require authentication before data access.
- The system shall protect data from unauthorized access.

Reliability

- The system shall prevent data loss during crashes.
- The system shall automatically save entries.

Compatibility

- The system shall run on Android 8.0 and above.
- The system shall be developed using Android Studio.

Technical Requirements

Development Tools

- Android Studio IDE
- Java or Kotlin programming language

Database

- SQLite or Room Database (local storage)

Architecture

- MVVM (Model–View–ViewModel)

Libraries

- MPAndroidChart (for graphs)
- Firebase Authentication (optional)
- Firebase Cloud Firestore (optional)

Constraints

- Must operate offline for core features.
- Must not require paid services.
- Must be completed within the project timeline.

Assumptions

- Users have basic smartphone literacy.
- Users will manually input income and expenses.
- Internet connection is only required for cloud sync and backup.