FINAL PROJECT REPORT

1. Introduction

The BudgetingApp is a mobile-based personal financial management system developed to help users track income, expenses, budgets, and savings goals efficiently. The application is designed for Android devices and provides both offline functionality and optional cloud synchronization.

The purpose of this project is to develop a secure, user-friendly, and reliable budgeting system that enables users to monitor their financial activities in real time. The system integrates structured data storage, automated calculations, and graphical reports to enhance financial decision-making.

2. Primary Requirements

Functional Requirements

User Account Management
- Allow users to register using email and password.
- Allow secure login and logout.
- Allow password reset functionality.
- Store user profile details such as name, email, and preferred currency.

Income Management
- Add, edit, and delete income records.
- Store income amount, source, and date.
- Display total income per selected period.

Expense Management
- Add, edit, and delete expense records.
- Store expense amount, category, description, and date.
- Support quick-add functionality and optional notes.

Expense Categorization

- Provide predefined categories (Food, Transport, Rent, Utilities, Entertainment, Education, Health).
- Allow custom category creation, editing, and deletion.

Budget Management
- Create weekly or monthly budgets per category.
- Display remaining budget.
- Notify users when approaching or exceeding limits.

Financial Goals
- Create savings goals with target and deadline.
- Display goal progress using:
  Progress (%) = (Saved Amount / Target Amount) × 100

Reports and Visualization
- Generate category spending reports.
- Display pie charts, bar charts, and line graphs.
- Allow filtering by date range.

Notifications and Backup
- Send expense reminders and budget alerts.
- Store data locally with optional cloud sync.
- Allow export as CSV or PDF.

Non-Functional Requirements

Usability
- Simple interface with light/dark mode.
- Maximum three taps to add expense.

Performance
- Load within 3 seconds.
- Handle at least 10,000 records.

Security
- Encrypt passwords.
- Require authentication before access.

Reliability
- Prevent data loss.
- Auto-save entries.

Compatibility
- Run on Android 8.0 and above.
- Developed using Android Studio.

3. System Operation

Architecture
The application follows the MVVM (Model–View–ViewModel) architecture:
- Model: Handles data and business logic.
- View: User interface.
- ViewModel: Connects Model and View.

User Authentication Process
- Validate user credentials.
- Encrypt passwords.
- Redirect to dashboard upon successful login.

Dashboard
- Displays total income, expenses, and balance.
- Updates automatically after transactions.

Income and Expense Processing
- Data entered via forms.
- Validation performed.
- Stored in local database.
- Real-time recalculation of totals.

Budget Monitoring
Remaining Budget = Set Budget – Total Category Expenses
- 80% threshold triggers warning.
- Exceeding limit triggers alert.

Savings Tracking
- Progress bar displays real-time goal progress.

Reports
- Pie charts for category distribution.
- Bar charts for income vs expenses.
- Line graphs for financial trends.

4. Technology Justification

Development Environment
- Android Studio for professional development and debugging.

Programming Language
- Kotlin (preferred) or Java.
- Kotlin offers null safety and concise syntax.

Database
- SQLite or Room Database for offline storage and fast access.

Architecture Choice
- MVVM ensures scalability, maintainability, and separation of concerns.

Security
- Encrypted passwords and input validation.

Data Visualization
- MPAndroidChart library for graphical representation.

Real-Time Processing
- Immediate recalculations for improved user experience.

Scalability
- Supports future upgrades like cloud sync and analytics.

5. Conclusion

The BudgetingApp successfully meets all functional and non-functional requirements. It provides structured financial tracking, real-time updates, secure authentication, and meaningful visual reports. The chosen technologies ensure efficiency, reliability, scalability, and strong user experience.