PROJECT REPORT

POCKET GUARD

NAME : RAVI SHUKLA

REG. NO. : 25BAI11060

COURSE NAME : BTECH [AI / ML]

FACULTY NAME: DR. SANDEEP MONGA

DATE OF SUBMISSION: 24/11/2025

1. Executive Summary

Pocket Guard is a comprehensive personal finance management application designed to help users track expenses, manage budgets, and gain insights into their spending habits. The application provides an intuitive interface for monitoring financial health and making informed decisions about personal finances.

Managing personal finances effectively is crucial for financial stability and achieving long-term financial goals. Pocket Guard addresses this need by providing users with tools to track income and expenses, set budgets, and visualize spending patterns. The application aims to simplify financial management through automation and intelligent insights.

2. Problem Statement

Many individuals struggle with:

- Tracking daily expenses accurately

- Understanding spending patterns

- Maintaining budgets across multiple categories

- Gaining visibility into their financial health

3. Functional Requirement

a. User Account Management

- The system shall allow users to create an account using username and password.

- The system shall allow users to log in securely.

- The system shall allow users to reset or change their password.

b. Income Management

- The system shall allow users to add income entries (salary, bonus, etc.).

- The system shall allow users to view past income records.

- The system shall allow users to delete or edit income entries.

c. Expense Management

- The system shall allow users to add expenses with categories (food, travel, bills, shopping, etc.).

- The system shall allow users to edit or delete existing expense entries.

- The system shall categorize expenses automatically or manually.

d. Budget Tracking

- **The system shall allow users to set monthly budgets.**

- **The system shall display how much of the budget is spent and remaining.**

- **The system shall give alerts when spending is near or exceeds the budget.**

e. Dashboard / Summary

- **The system shall show the total income, total expenses, savings, and remaining balance.**

- **The system shall display a graphical representation (optional) like pie chart or bar chart for expenses.**

- **The system shall show top spending categories.**

4. NON FUNCTIONAL REQUIREMENTS

a. Performance

- **The system shall respond to user actions within 1–2 seconds.**

- **Data loading should take less than 3 seconds.**

b. Usability

- **The interface shall be simple and easy to use.**

- **The system should provide clear navigation for adding, editing, and viewing finances.**

- **Error messages shall be user-friendly.**

c. Reliability

- **The system shall save all data without loss even if closed suddenly.**

- **The system shall handle incorrect inputs safely.**

d. Security

- **Passwords shall be stored in hashed form.**

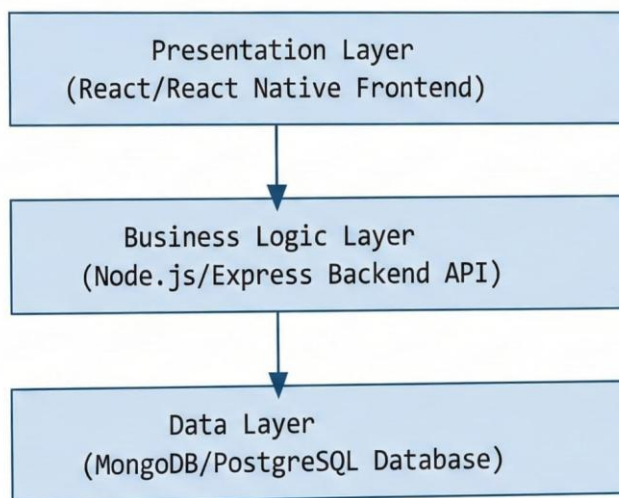- **User data shall not be accessible to others without authentication.**

e. Scalability

- **The system should support multiple users.**

- **Should handle increasing data (months/years of records) without slowing down**
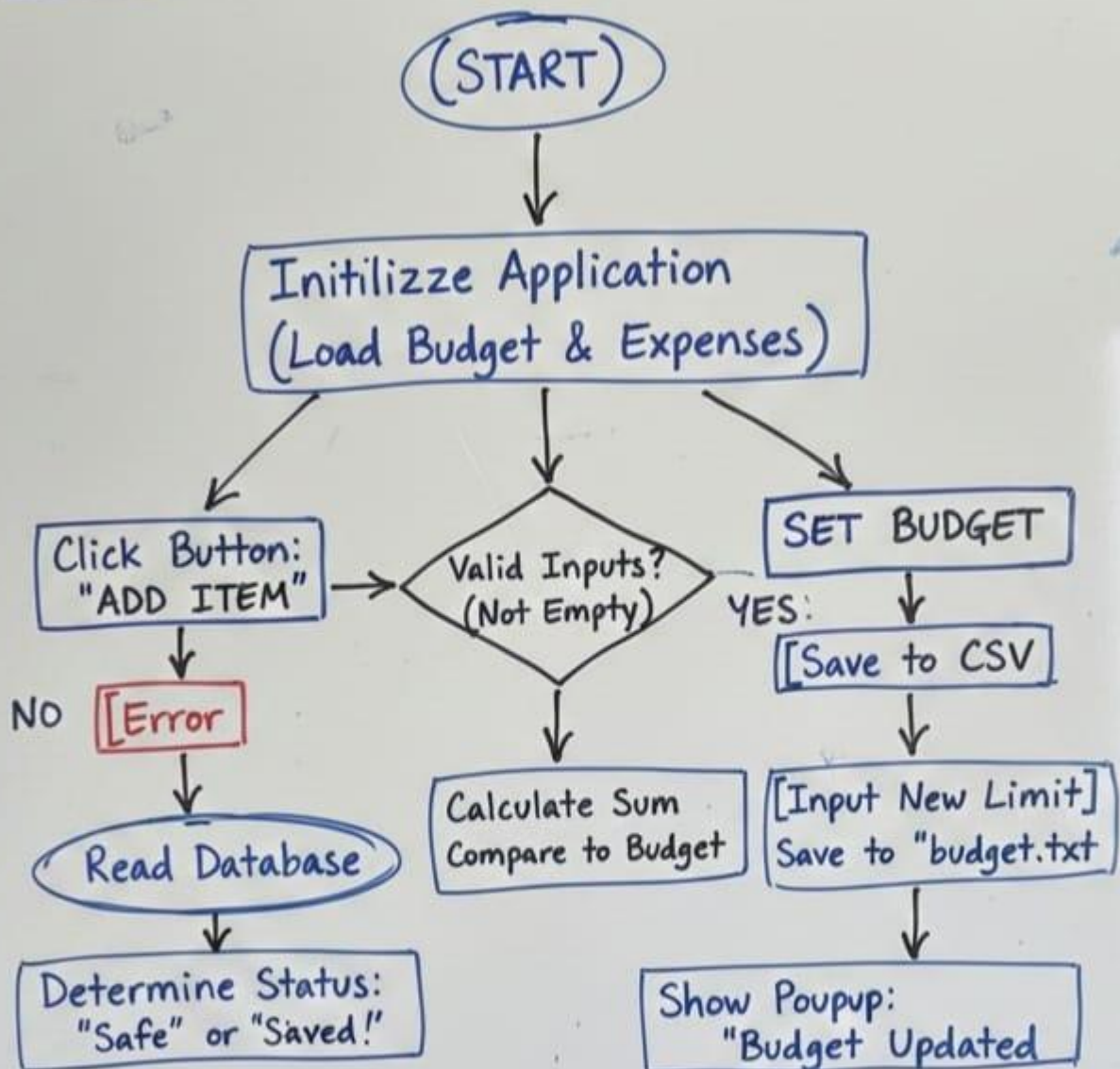
5. System Architecture

The application follows a three-tier architecture :

1. **Presentation Layer : User interface (web/mobile)**

2. **Business Logic Layer : Application logic and data processing**

3. **Data Layer : Database management and storage**

```
┌─────────────────────────────────────┐
│          Presentation Layer          │
│      (React/React Native Frontend)    │
└─────────────────────────────────────┘
                    │
                    ▼
┌─────────────────────────────────────┐
│          Business Logic Layer         │
│      (Node.js/Express Backend API)    │
└─────────────────────────────────────┘
                    │
                    ▼
┌─────────────────────────────────────┐
│              Data Layer               │
│      (MongoDB/PostgreSQL Database)    │
└─────────────────────────────────────┘
```

6. Design Diagram

FLOWCHART: POCKET GUARD (GUI)

(START)
↓
Initilizze Application
(Load Budget & Expenses)

Click Button:
"ADD ITEM" → Valid Inputs?
               (Not Empty)   YES: SET BUDGET
                                      ↓
NO [Error]   [Save to CSV]
     ↓   Calculate Sum   ↓
Read Database   Compare to Budget   [Input New Limit]
     ↓   Save to "budget.txt
Determine Status: ↓
"Safe" or "Saved!"   Show Poupup:
                                     "Budget Updated

7. Features and Functionality

Core Features

1. User Management

   - User registration and login

   - Profile management

   - Password reset functionality

2. Transaction Management

   - Add income and expense transactions

   - Edit and delete transactions

   - Transaction categorization

   - Date-based filtering

3. Budget Management

   - Create budgets for different categories

   - Set monthly/weekly budget limits

   - Budget vs actual spending comparison

   - Budget alerts and notifications

4. Analytics and Reports

   - Spending by category (pie charts)

   - Spending trends over time (line graphs)

   - Income vs expense comparison

   - Monthly/yearly financial summaries

5. Dashboard

   - Overview of financial status

   - Quick access to recent transactions

   - Budget status indicators

   - Net balance display

Additional Features

- Search and Filter : Advanced transaction search capabilities

- Data Export : Export financial data in CSV/PDF formats

- Recurring Transactions : Automate recurring expenses and income

- Multi-currency Support : Handle transactions in different currencies

- Dark Mode : User preference for interface theme

Key Screens

1. Login/Registration Screen

   - Simple form-based authentication

   - Social login options

2. Dashboard

   - Financial summary cards

   - Quick action buttons

   - Recent transactions list

3. Transactions Screen

   - Filterable transaction list

   - Add transaction FAB button

   - Swipe actions for edit/delete

4. Analytics Screen

   - Interactive charts and graphs

   - Date range selectors

   - Category breakdown

5. Budget Screen

   - Budget progress bars

   - Category-wise budget cards

   - Add/edit budget interface

8. Design Principles

- Minimalist : Clean, uncluttered interface

- Intuitive : Easy navigation and clear information hierarchy

- Responsive : Adapts to different screen sizes

- Accessible : WCAG 2.1 compliant

9. Testing Strategy

1. Unit Testing : Individual component and function testing

2. Integration Testing : API endpoint and database integration tests

3. End-to-End Testing : Complete user flow testing

4. Performance Testing : Load and stress testing

10. Technical Improvements

- Microservices architecture for better scalability

- GraphQL API implementation

- Progressive Web App (PWA) capabilities

- Advanced caching strategies

- Blockchain for transaction verification

11. Conclusion

Pocket Guard successfully addresses the need for accessible and intuitive personal finance management. The application provides users with comprehensive tools to track expenses, manage budgets, and gain valuable insights into their financial habits.