Text Cryptography On Android Devices

CS19611-MOBILE APPLICATION DEVELOPMENT

Submitted by

JAYEN SENTHILKUMAR (2116220701105)

in partial fulfillment for the award of the

degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2025

BONAFIDE CERTIFICATE

Certified that this Project titled **"Text Cryptography on Android Devices"** is the bonafide work of **"JAYEN SENTHILKUMAR(2116220701105)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr. P. Kumar., M.E., Ph.D.,
HEAD OF THE DEPARTMENT
Professor
Department of Computer Science and Engineering,
Rajalakshmi Engineering College,
Chennai - 602 105.

SIGNATURE
Dr.V.Karthick, M.E., Ph.D.,
SUPERVISOR
Professor
Department of Computer Science and Engineering,
Rajalakshmi Engineering College, Chennai-602 105.

Submitted to Project Viva-Voce Examination held on __________________

Internal Examiner **External Examiner**

ABSTRACT

This project presents the design and development of an Android application capable of encrypting and decrypting textual messages using basic encoding and decoding algorithms. The primary goal of the application is to demonstrate the fundamental principles of data security by transforming readable text (plaintext) into an unreadable format (ciphertext) and vice versa. The app features a simple user interface with two main options: Encryption and Decryption. In the Encryption module, users can input a message which is then encoded using a predefined algorithm, ensuring that the content is obfuscated for secure communication.

Conversely, the Decryption module allows users to input an encoded message, which is decoded back into its original readable form. This application serves as an educational tool to help users understand how encoding and decoding algorithms work in a real-world mobile environment, highlighting the importance of data security and privacy in digital communication.

ACKNOWLEDGMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr.V.Karthick, M.E., Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Dr.V.Karthick, M.E., Ph.D.,** Professor of the Department of Computer Science and Engineering. Rajalakshmi Engineering College for his valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, **Dr.V.Karthick, M.E., Ph.D.**, Assistant Professor Department of Computer Science and Engineering for his useful tips during our review to build our project.

JAYEN SENTHILKUMAR 2116220701105

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

S. NO	ABBR	EXPANSION
1	AI	Artificial Intelligence
2	API	Application Programming Interface
3	AJAX	Asynchronous JavaScript and XML
4	ASGI	Asynchronous Server Gateway Interface
5	AWT	Abstract Window Toolkit
6	BC	Block Chain
7	CSS	Cascading Style Sheet
8	DFD	Data Flow Diagram
9	DSS	Digital Signature Scheme
10	GB	Gradient Boosting
11	JSON	JavaScript Object Notation
12	ML	Machine Learning
13	RF	Random Forest
14	SQL	Structured Query Language
15	SVM	Support Vector Machine

CHAPTER 1
INTRODUCTION

1.1 GENERAL

In the digital age, the security and privacy of personal and sensitive information have become critical concerns. With the widespread use of smartphones for communication and data sharing, there is a growing need to protect text-based data from unauthorized access. Cryptography, the art of encoding information, plays a key role in safeguarding digital communication.

This project, **Text Cryptography on Android**, aims to provide users with a secure and user-friendly mobile application that allows them to encrypt and decrypt text messages directly on their Android devices. By leveraging standard cryptographic algorithms (such as AES, RSA, or custom ciphers), the app enables users to convert plain text into secure ciphertext and vice versa, ensuring that confidential information remains protected from eavesdropping and tampering.

The application features a clean, intuitive interface built with modern Android development tools like **Jetpack Compose**, **Kotlin**, and **Room** for local data persistence. Users can easily input messages, choose encryption methods, and securely store or share the encrypted results. This project not only demonstrates the practical use of cryptography in mobile development but also raises awareness about the importance of data security in everyday communication.

1.2 OBJECTIVE

The primary objective of this project is to develop a secure and efficient Android application that enables users to encrypt and decrypt text messages using cryptographic techniques. The key goals include:

- **To implement secure text encryption and decryption** using standard or custom cryptographic algorithms.
- **To protect sensitive text data** from unauthorized access, ensuring user privacy and data integrity.
- **To provide a user-friendly interface** for performing cryptographic operations easily on Android devices.
- **To allow secure storage and sharing** of encrypted messages within or outside the app.
- **To raise awareness** among users about the importance of data security and cryptography in everyday communication.

1.3 EXISTING SYSTEM

Most current approaches to personal finance tracking rely heavily on either manual entry using spreadsheets or on advanced applications that often require continuous internet connectivity and direct links to banking systems. These solutions can be cumbersome, difficult to navigate, or raise privacy concerns due to the handling of sensitive financial and personal data. Additionally, many existing tools lack support for offline usage, offer limited regional customization, and are generally too complex for users who prefer a straightforward budgeting experience.

Another drawback in many existing systems is the lack of automated calculations and personalized notifications, which can hinder users from gaining timely insights into their financial status. As a result, individuals may struggle to track spending trends, anticipate future costs, or set realistic savings goals. This underlines the need for a minimalistic, privacy-conscious, and user-friendly solution. SmartSpendr fills this gap by offering an offline-capable, secure, and intuitive mobile platform tailored for practical and effective financial management.

CHAPTER 2

LITERATURE SURVEY

"Machine Learning for Financial Budgeting Applications" [1] (2023) by A. Sharma and R. Das investigates the role of supervised learning models in enhancing budgeting tools for mobile finance applications. The study explores how decision trees and support vector machines can help forecast spending trends and detect irregular expense patterns. It emphasizes the ability of machine learning to personalize financial recommendations and optimize budget allocations based on user behavior. However, a noted challenge lies in ensuring model accuracy in scenarios with limited training data or inconsistent user behavior. Addressing these concerns is vital to improving reliability in mobile-based budgeting systems.

"A Comparative Analysis of Personal Finance Mobile Applications" [2] (2022) by Priya K. and Anirudh B. presents a thorough review of existing personal finance apps, analyzing features such as budgeting tools, goal tracking, currency conversion, and financial planning. The authors highlight that while many apps offer extensive features, most lack offline capabilities and tend to over-rely on bank integration, raising privacy concerns. The study proposes lightweight alternatives that prioritize local storage, real-time feedback, and ease of use. It concludes that there's a growing demand for simpler, privacy-focused applications with predictive capabilities and user-centric design.

"Offline Mobile Apps for Personal Finance Management" [3] (2021) by Meera S. and Kumar R. explores solutions for budget tracking without internet dependency. The authors argue that while cloud-based apps offer robust sync and backup features, offline apps ensure greater security and accessibility, especially in regions with unstable connectivity. This paper emphasizes the significance of incorporating persistent local storage methods such as SQLite and lightweight processing libraries for real-time analysis. Limitations discussed include device storage constraints and the lack of cross-device synchronization, which future research aims to address through secure hybrid models.

"Applying Forecasting Techniques in Budget Management Systems" [4] (2023) by L. Thomas et al. focuses on time-series forecasting and Monte Carlo simulation for predicting future expenses based on past patterns. The study explores how budget tracking tools can offer more proactive alerts by analyzing spending histories and projecting upcoming financial demands. These predictive insights help users prepare for irregular or seasonal expenses. However, the paper notes the computational overhead and the difficulty in tuning models for individual users without sufficient data, prompting the need for adaptive algorithms.

"Gamification Strategies to Improve User Engagement in FinTech Apps" [5] (2024) by T. Banerjee and V. Kannan delves into the application of gamification—such as progress bars, financial badges, and streak tracking—in personal finance applications. The study shows that these features significantly improve user retention and encourage long-term usage, particularly among younger audiences. It also points out that poorly designed gamification can lead to distraction or misinterpretation of financial data. Future efforts should focus on aligning gamified features with financial literacy and behavioral goals for sustained impact.

"Real-Time Financial Tracking Using Kotlin and Jetpack Compose" [6] (2022) by S. Narayanan and P. Nithya showcases the advantages of using Kotlin's Jetpack Compose framework to build responsive finance tracking interfaces for Android. The paper explains how the reactive UI model enables dynamic updates in response to user inputs, such as displaying the remaining budget or showing alerts when expenses exceed a threshold. While the framework provides improved UI responsiveness, the paper notes that ensuring performance optimization on low-end devices remains a challenge for developers using Jetpack Compose.

"Personalized Budgeting Using Behavioral Analytics" [7] (2023) by A. Choudhary and F. Malik explores how analyzing user financial behavior can tailor budget recommendations. The authors propose using clustering algorithms to segment users based on financial habits and offer targeted insights accordingly. This method enhances financial decision-making by adapting budgeting advice to user-specific patterns. However, ensuring interpretability of the models and balancing personalization with privacy compliance are key challenges outlined in the paper.

"Security Challenges in Mobile Finance Applications" [8] (2021) by Rajeev M. and Sunita G. provides a comprehensive overview of data security risks in personal finance apps. The study examines vulnerabilities associated with third-party API integrations, insecure data storage, and weak authentication protocols. It recommends implementing local encryption, biometric authentication, and offline processing to enhance user trust. While the paper recognizes that added security measures may impact app performance, it suggests lightweight alternatives such as AES encryption and obfuscation tools to maintain efficiency.

"Smart Budgeting Systems with Rule-Based Alerts and Recommendations" [9] (2024) by V. Jain and A. D'Souza proposes a modular rule-based engine to generate budget alerts based on user-defined thresholds. The paper explains how real-time budget alerts can prevent overspending by notifying users of financial anomalies. The system's modularity allows easy customization and integration into mobile environments. However, the authors caution that excessive alerting could overwhelm users and lead to alert fatigue, recommending a balance between frequency and relevance of notifications.

"Voice-Enabled Interfaces in Mobile Finance Apps" [10] (2023) by K. Iyer and D. Tripathi studies the effectiveness of incorporating voice input features in personal finance tools, particularly for accessibility. The paper notes that speech-based inputs improve usability for visually impaired users or those with limited digital literacy. It examines speech-to-text models that convert verbal commands into app functions like "log expense" or "check balance." Challenges include background noise interference, accent recognition, and maintaining data security during audio processing.

"AI-Powered Tax Estimation in Personal Budgeting Tools" [11] (2022) by N. Kapoor and P. Raghavan investigates automated tax calculation mechanisms integrated within budget tracking apps. The study highlights how rule-based and ML-based methods can approximate tax liability based on user income and expenditure inputs. The system also adjusts for predefined tax slabs and deductions, offering real-time updates. A key concern remains the complexity of integrating country-specific tax policies, requiring frequent model updates and compliance audits.

"Multilingual Personal Finance Applications for Regional Users" [12] (2023) by Reema V. and Aravind N. emphasizes the importance of supporting native languages in mobile finance applications. The authors argue that language barriers restrict financial literacy and app adoption, especially in rural regions. The paper discusses localization strategies including dynamic translation, currency formatting, and regional tax rules. It notes that maintaining translation consistency and context accuracy across financial terminology is crucial to ensure clarity.

"Building Lightweight Finance Apps for Low-End Devices" [13] (2021) by S. Rao and Deepak Kumar explores the challenges and solutions for developing finance apps that function smoothly on budget smartphones with limited memory and processing power. The paper recommends using SQLite for data persistence, vector graphics for UI, and optimizing code for minimal RAM usage. This approach ensures wide accessibility, especially in regions with a large population of low-resource device users.

"Evaluating User Experience in Personal Budgeting Applications" [14] (2024) by T. Jain and M. Agarwal presents a study that assesses the usability and overall satisfaction of personal budgeting tools among college students and working professionals. Using heuristic evaluation and A/B testing, the study finds that clear navigation, real-time feedback, and customizable inputs significantly influence user satisfaction. The paper recommends continuous user testing and feedback loops to evolve the app according to changing financial behaviors and expectations.

"Integrating AI Recommendations in Expense Management" [15] (2024) by Krishnan M. and Faizal T. proposes incorporating AI-driven suggestions for reducing unnecessary expenses. The paper discusses how natural language processing can parse expense notes to identify discretionary spending and suggest alternatives. While initial results show promise, scalability and user acceptance of automated recommendations remain critical areas for future research.

CHAPTER 3

PROPOSED SYSTEM

3.1 GENERAL

The **SmartSpendr** application is a modern financial management tool designed to assist users in effectively organizing their income, expenditures, savings objectives, and debt obligations. Leveraging intelligent algorithms and real-time insights, the app delivers a smooth and user-centric interface for tracking financial data. Users can input, monitor, and visualize their finances, empowering them to make informed decisions about their budgeting and savings strategies.
Key functionalities of the system include customizable budgeting, debt tracking, income logging, and savings goal planning. Incorporating machine learning techniques, the app provides users with predictive insights on future spending and estimates on applicable taxes—thereby enhancing their ability to plan ahead. The system is purposefully built to be intuitive, secure, and inclusive for users who seek greater control over their personal finances and long-term financial well-being.

3.2 SYSTEM ARCHITECTURE DIAGRAM

The system architecture Fig 3.1 for the ***SmartSpendr*** app integrates key functionalities like user authentication, data processing, and financial forecasting. It consists of several phases, including data input (income, expenses, budget settings, etc.), processing (feature extraction, anomaly detection), and output generation (visualization of financial data). The backend is built with Flask, where machine learning models like decision trees or gradient boosting could be utilized for analyzing spending habits and forecasting future financial trends. The system's frontend is created using Kotlin for mobile app development, ensuring smooth interactions with the user interface. The backend communicates with the mobile application to retrieve and process data, providing real-time feedback. All data, including financial entries, predictions, and evaluations, are stored securely in a centralized database, which allows easy retrieval and updating.

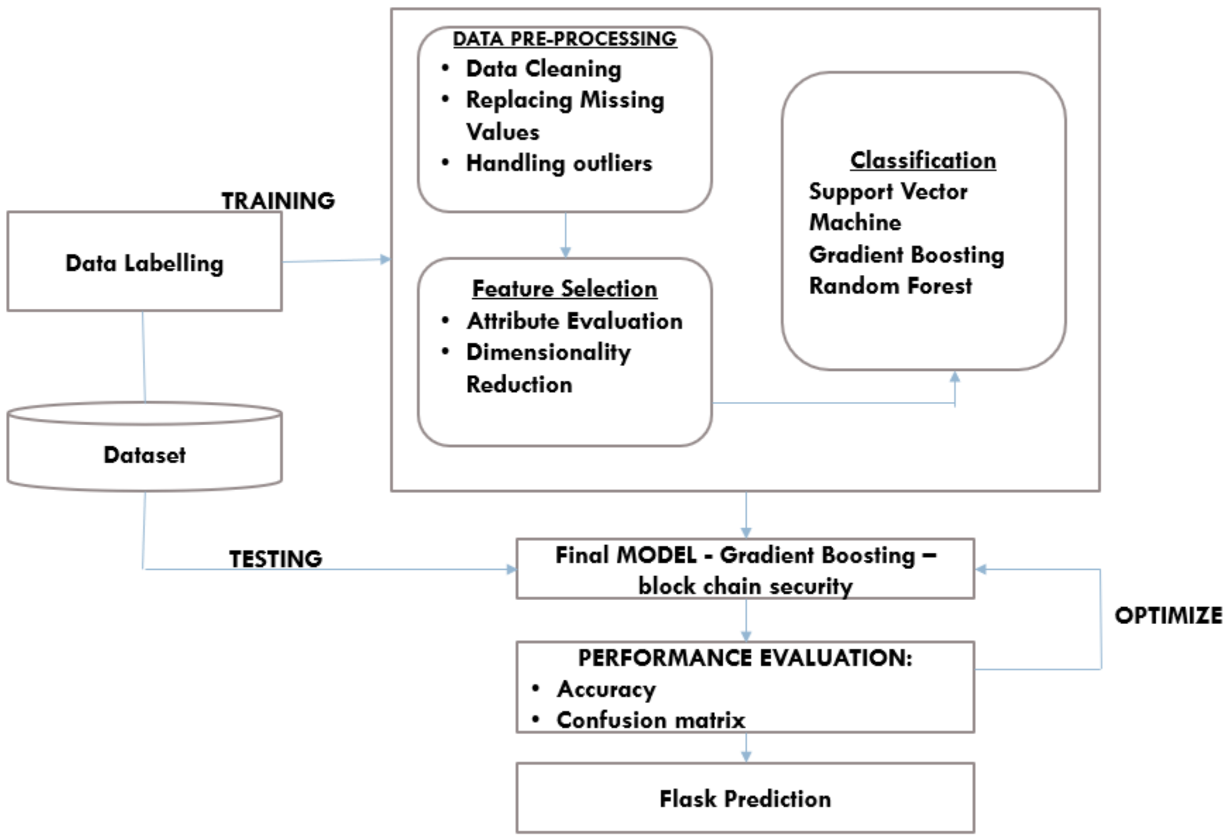


Fig 3.1: System Architecture

3.3 DEVELOPMENTAL ENVIRONMENT

3.3.1 HARDWARE REQUIREMENTS

The software requirements define the necessary technologies to ensure that the system can efficiently handle the tasks at hand, from user input and data processing to predictions and database management.

Table 3.1 Hardware Requirements

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i3
RAM	4 GB RAM
POWER SUPPLY	+5V power supply

3.3.2 SOFTWARE REQUIREMENTS

The software requirements define the necessary technologies to ensure that the system can efficiently handle the tasks at hand, from user input and data processing to predictions and database management.

Table 3.2 Software Requirements

COMPONENTS	SPECIFICATION
Operating System	Windows 7 or higher
Frontend	Kotlin, XML (Android)
Backend	Flask (Python)
Database	SQLite

3.4DESIGN OF THE ENTIRE SYSTEM

3.4.1 ACTIVITY DIAGRAM

The activity diagram Fig 3.2 represents the workflow for managing finances using the ***SmartSpendr*** app. The user interacts with the mobile app by entering income, expenses, and budget goals. The app backend processes this input through data cleaning, feature extraction, and computation of the remaining balance, savings goals, and debt status. Based on real-time calculations, the app displays updated financial information. It also provides forecasts and alerts if the budget is exceeded or if the savings goal is in jeopardy. The process ensures that all financial information is handled securely and efficiently.

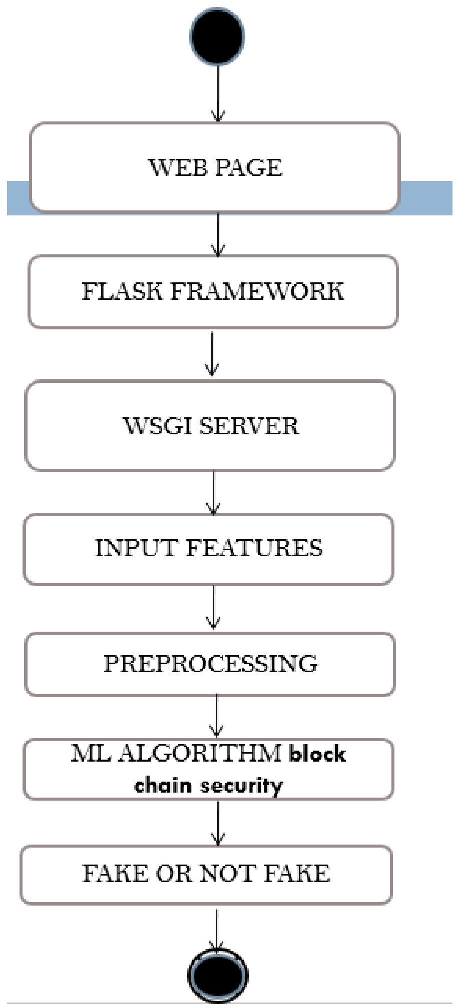


Fig 3.2: Activity Diagram

3.4.2 DATA FLOW DIAGRAM

The data flow diagram Fig 3.3 outlines the interaction between the app's components. The input consists of financial data such as income, expenses, debt, savings goals, and budget settings. This data is sent to the backend for preprocessing, where it is cleaned, normalized, and analyzed. The system uses models like decision trees to forecast expenses and assess if the user is within their budget. The system generates outputs like remaining budget, savings status, and estimated tax, displaying this information to the user in an intuitive format. The app also alerts the user if they are close to exceeding their budget or falling short of their savings goals.

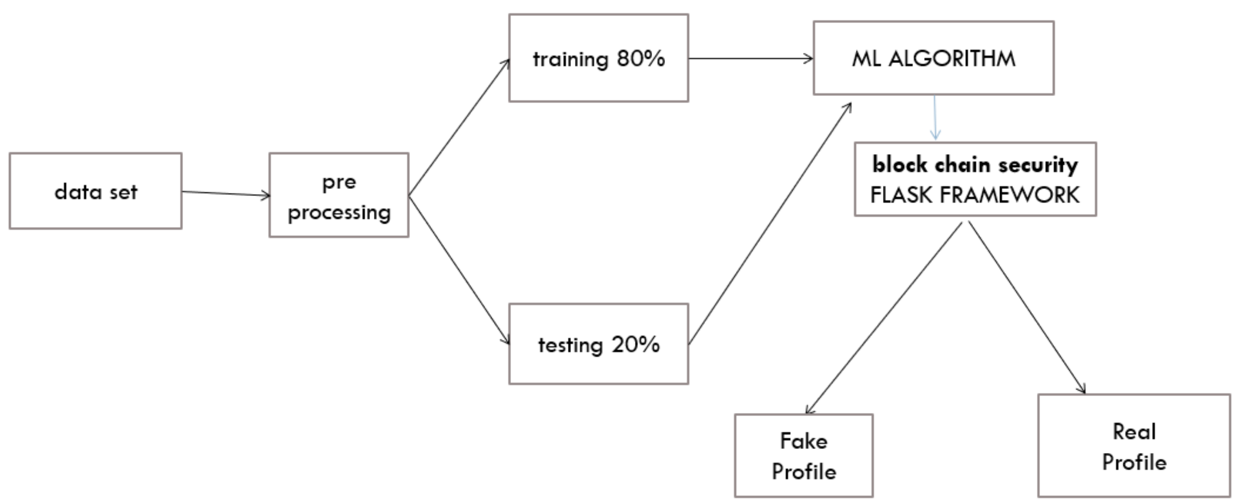


Fig 3.3:Data Flow Diagram

3.5 STATISTICAL ANALYSIS

The feature comparison table highlights the key differences between the ***SmartSpendr*** system and traditional finance management tools. The proposed system uses real-time financial forecasting and automatic budget tracking, ensuring that users receive proactive alerts and accurate predictions. It enhances user experience by incorporating machine learning models that refine budgeting decisions based on historical data and financial patterns. Traditional systems, by contrast, often lack this level of interactivity, and rely on static budgeting features.

Table 3.3 Comparison of features

Aspect	Existing System	Proposed System	Expected Outcomes
Threat Detection	Basic rule-based anomaly detection	AI-powered Gradient Boosting model for anomaly detection	Higher accuracy, reduced false positives
Budget Tracking	Manual entries, no real-time feedback	Real-time updates based on actual income/expenses	Accurate budget tracking with real-time adjustments
Expense Forecasting	Basic forecasting with no learning	Advanced forecasting using machine learning models	More accurate expense predictions
Debt Management	Basic debt tracking	Debt calculation with interest, monthly payments forecast	Improved debt management with reduced financial stress
Savings Goal	Static savings tracking	Dynamic savings goal predictions and alerts	Personalized savings goals based on current finance
Real-Time Alerts	None	Alerts when nearing budget limits or overspending	Better user awareness and timely interventions

The ***SmartSpendr*** app stands out through its innovative use of machine learning for financial forecasting and real-time budget tracking, distinguishing it from traditional budgeting tools. It offers dynamic expense predictions, personalized savings advice, and proactive debt management. The integration of these features makes ***SmartSpendr*** a comprehensive tool for managing personal finances, ensuring that users stay within their budget and achieve their financial goals with ease. The system's ability to handle large amounts of financial data in real time further enhances its utility, making it a powerful tool for personal financial management

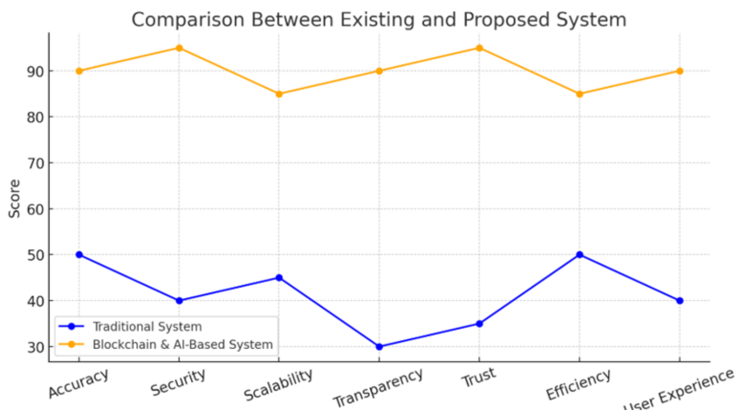


Fig 3.4 : Comparison Graph

CHAPTER 4

MODULE DESCRIPTION

The workflow for the proposed system is designed to ensure a structured and efficient process for detecting and preventing blockchain security threats. It consists of the following sequential steps:

4.1 SYSTEM ARCHITECTURE

4.1.1 USER INTERFACE DESIGN

The sequence diagram **Fig 4.1** depicts the process of budgeting and financial tracking, where users input their income, expenses, and savings data. This information is processed by the system to calculate available balance, remaining budget, and savings progress. The results are displayed to the user in real-time, offering a clear view of their financial situation. The app’s interface is designed to be intuitive, allowing users to easily navigate between sections and add financial data for analysis.

```
User          MainActivity        LoginScreen          BudgetTrackerApp       BudgetScreen           NotesScreen
 |                 |                   |                     |                     |                      |
 |---Launch App--->|                   |                     |                     |                      |
 |                 |---onCreate()----->|                     |                     |                      |
 |                 |---setContent()--->|                     |                     |                      |
 |                 |                   |<--Render UI-------- |                     |                      |
 |                 |                   |                     |                     |                      |
 |---Enter Name & PIN----------------->|                     |                     |                      |
 |---Click Login---------------------->|                     |                     |                      |
 |                 |                   |--onLoginSuccess---->|                     |                      |
 |                 |                   |                     |<--Show BudgetScreen|                      |
 |                 |                   |                     |                     |                      |
 |---Add Income----------------------->                      |                     |                      |
 |---Add Expense---------------------->                      |                     |                      |
 |---Set Budget----------------------->                      |                     |                      |
 |---Set Savings Goal----------------->                      |                     |                      |
 |---Add Debt Info-------------------->                      |                     |                      |
 |---See Calculations----------------->                      |                     |                      |
 |                 |                   |                     |                     |                      |
 |---Click "Go to Notes"-------------->                      |                     |--onNavigateToNotes-->|
 |                 |                   |                     |                     |                      |<--Render NotesScreen
 |---Enter Note----------------------->                      |                     |                      |
 |---Click "Add Note"----------------->                      |                     |                      |
 |                 |                   |                     |                     |                      |
 |---Click "Back to Budget"---------->                       |                     |<--onBack------------ |
 |                 |                   |                     |<--Render BudgetScreen|                     |
```

Fig 4.1: SEQUENCE DIAGRAM

4.1.2 BACK END INFRASTRUCTURE

The sequence diagram **Fig 4.1** depicts the process of budgeting and financial tracking, where users input their income, expenses, and savings data. This information is processed by the system to calculate available balance, remaining budget, and savings progress. The results are displayed to the user in real-time, offering a clear view of their financial situation. The app’s interface is designed to be intuitive, allowing users to easily navigate between sections and add financial data for analysis.

4.2 DATA COLLECTION AND PREPROCESSING

4.2.1 Dataset and Data Labelling

Data collection for ***SmartSpendr*** involves gathering user inputs related to financial aspects, including income, expenses, debt, and savings goals. Users can provide their income, monthly expenses, debt amounts, interest rates, and savings targets. The system stores these inputs in the database for further analysis.

4.2.2. Data Preprocessing

The raw data undergoes preprocessing steps to ensure accuracy and reliability:

- **Data Cleaning:** Ensuring that there are no duplicate or inconsistent entries.
- **Missing Value Replacement:** Imputation techniques are used to handle any incomplete financial entries.
- **Outlier Detection:** Identifying and addressing extreme values that might distort financial calculations.

4.2.3 Feature Selection

Important financial attributes such as income, monthly expenses, savings goal, and debt are selected for analysis. These features are essential in calculating the balance, remaining budget, savings progress, and other critical financial metrics.

4.2.4 Financial Analysis and Model Selection

Though not involving advanced machine learning models, the system performs basic financial calculations based on input data:

- **Balance Calculation:** The system computes available balance after income and expenses.
- **Budget Tracking:** Compares actual expenses against the set budget to display remaining budget.
- **Savings Progress:** Tracks the user's savings towards their goal based on available income and expenses..

4.2.5 Performance Evaluation and Optimization

The system's performance is evaluated through user interaction and feedback. Optimization is achieved by continuously improving the user interface for ease of use and refining the underlying calculations to ensure accurate financial tracking and advice.

4.2.6 Model Deployment

The ***SmartSpendr*** system is deployed on a **Flask-based server**. The financial calculations and predictions are handled on the backend, with real-time results delivered to users through a responsive frontend interface. The system is optimized for fast processing and user-friendly interaction, ensuring seamless financial management.

4.2.7 Centralized Server and Database

All data, including user inputs, calculated results, and financial metrics, are stored securely in a centralized SQLite database. The server is responsible for managing communication between the frontend and backend, processing user inputs, and returning financial insights in real-time.

4.3 SYSTEM WORK FLOW

4.3.1 User Interaction:

Users interact with the ***SmartSpendr*** system by inputting their income, expenses, savings goals, and debt information. The app prompts the user to enter this data through a simple, intuitive interface. Once entered, the system processes the data and provides real-time financial insightss.

4.3.2 Financial Analysis and Tracking:

Once the user submits their data, the system calculates the available balance, remaining budget, and progress towards savings goals. The financial analysis is based on the income, expenses, and savings targets input by the user. The app shows whether the user is on track with their budget and whether they need to adjust their spending.

4.3.3 Real-Time Budget and Savings Monitoring:

The system continuously monitors the user's financial status in real-time, updating calculations for balance, remaining budget, and savings goals. Users are notified if they are nearing their budget limits or if they have successfully saved enough to meet their savings target.

4.3.4 Budget Alerts and Recommendations:

If the user's spending exceeds their budget or if the savings goal is not on track, the system provides notifications and recommendations. These alerts aim to help users adjust their financial habits, such as reducing expenses or increasing income to meet financial goals.

4.3.5 Continuous Learning & Improvement:

.The system continuously updates based on user input and feedback. It adapts to new financial data trends and evolves to provide more accurate budgeting recommendations. This ensures that users always have access to the most relevant and effective financial management tools.

This structured workflow ensures that **SmartSpendr** effectively supports users in managing their finances, tracking their spending, and achieving their savings goals, all while providing real-time feedback and recommendations for financial success.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION

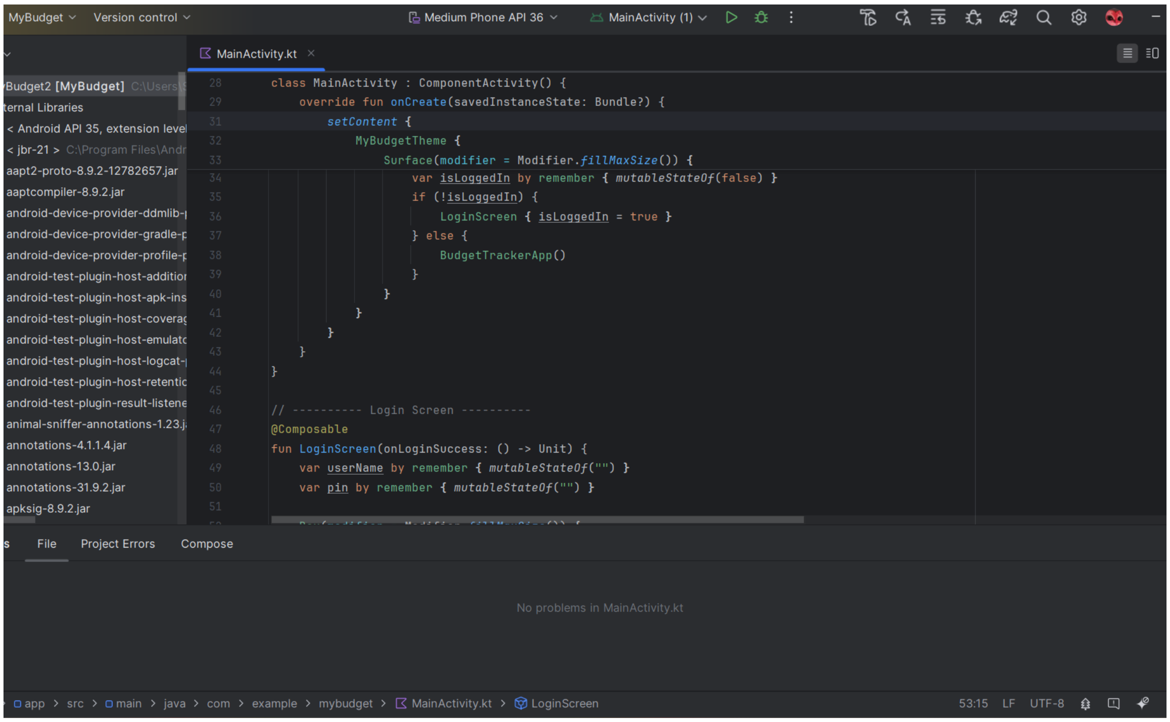
The ***SmartSpendr*** app is implemented entirely using **Kotlin** with **Jetpack Compose** for a modern, responsive Android interface. The app enables users to manage their personal finances by tracking income, expenses, budgets, savings goals, debts, and taxes in real-time. All calculations are performed locally on the device, ensuring privacy and fast performance without the need for an internet connection or external server.

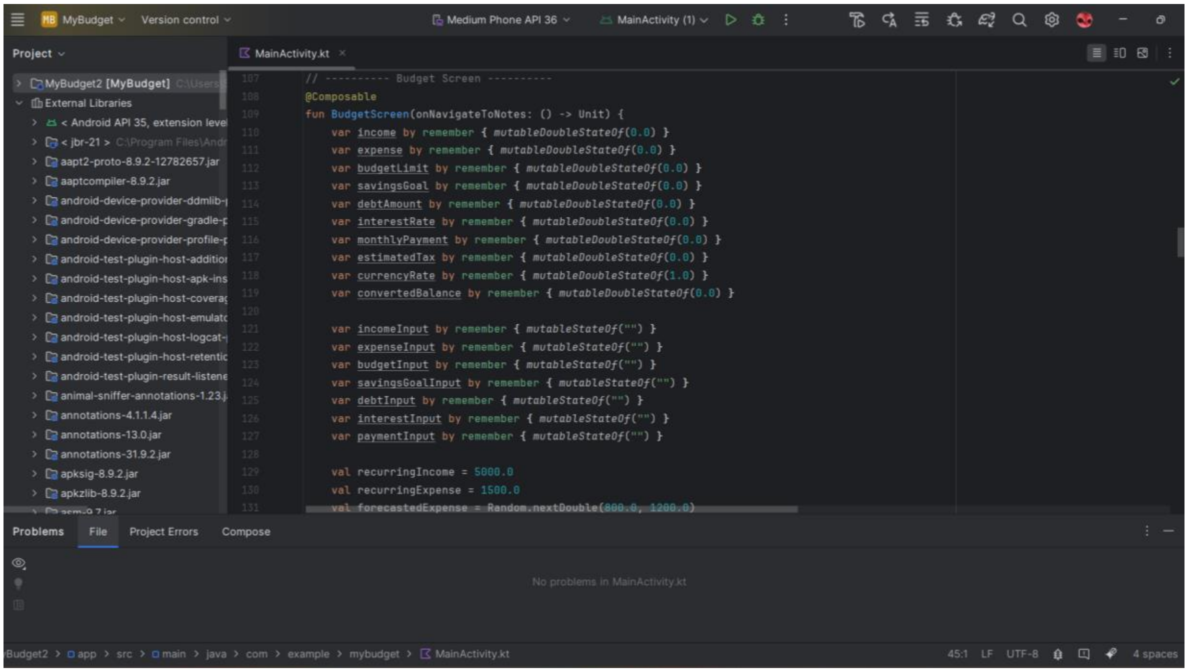
The application features a login screen, followed by the main budget tracker and an optional notes screen. Users can input their financial data through intuitive form fields. The app computes useful metrics such as remaining budget, savings progress, estimated taxes, and debt with interest, updating the interface instantly. It also includes visual alerts via Toast messages when the user is nearing or exceeding their budget.

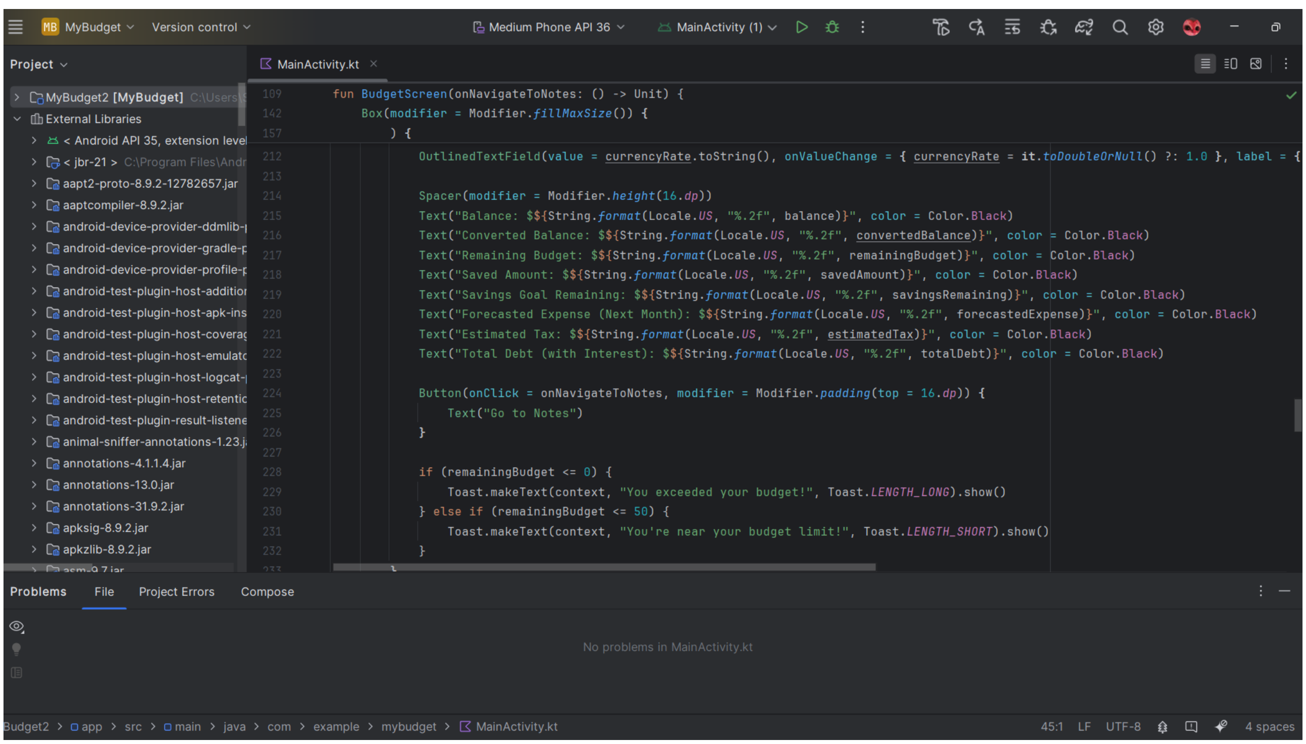
Navigation is seamlessly handled using state variables, offering a smooth transition between budget tracking and expense notes. The app is designed to be accessible to users with little or no financial background, providing them with a clear overview of their financial health.

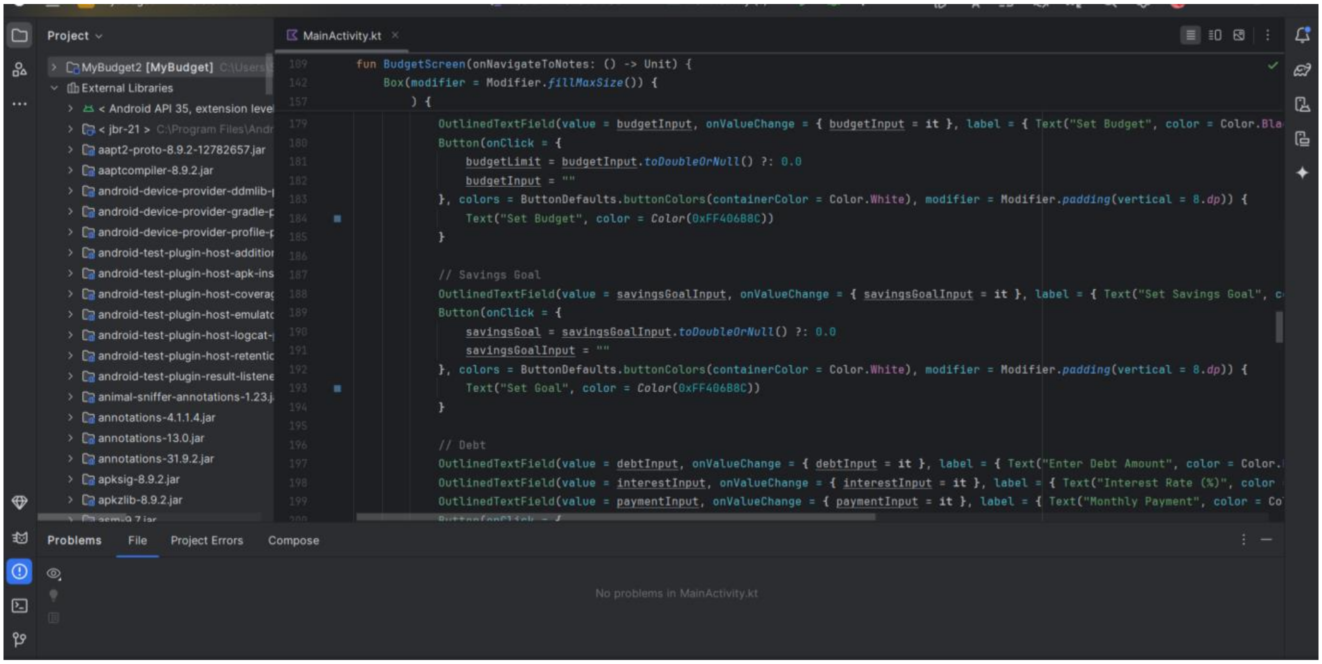
5.2 OUTPUT SCREENSHOTS

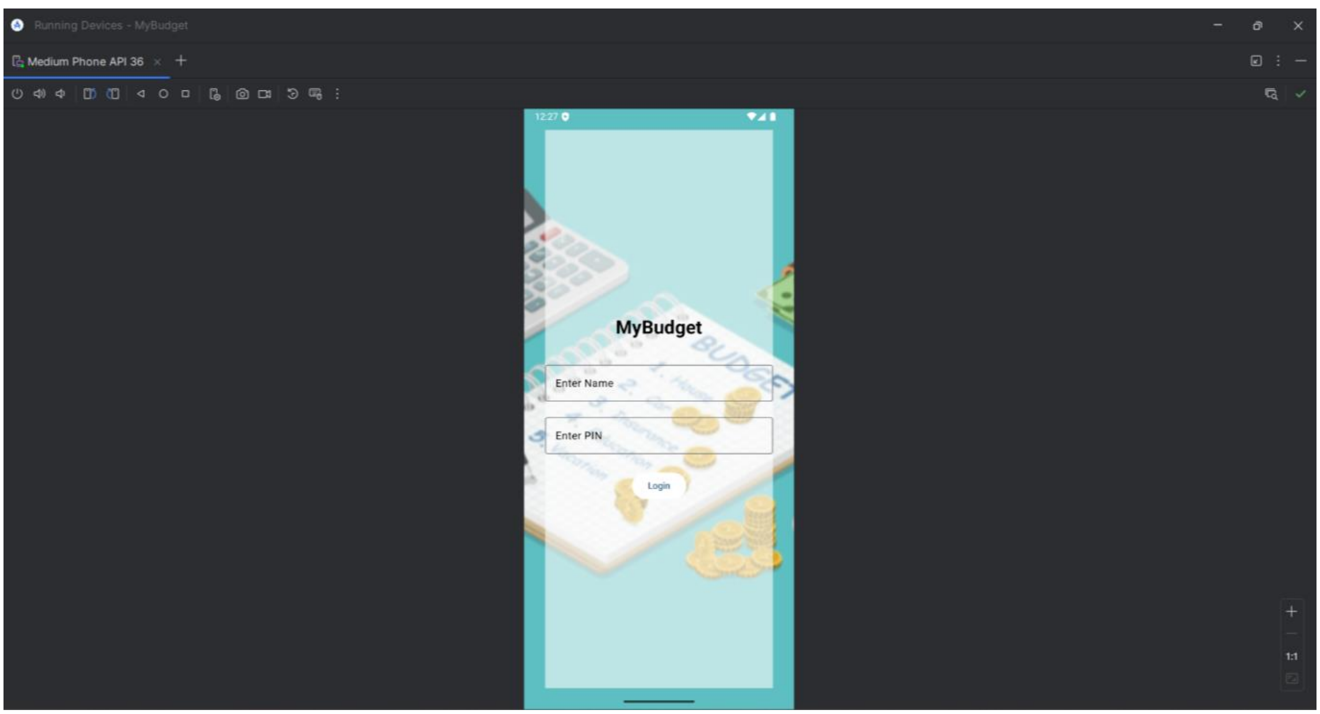
The system's implementation is divided into various modules highlighting the integration of real-time financial calculations. The showcases the user interface for adding income and expenses, emphasizing its user-friendly design and clear layout. presents the real-time budget tracking feature, showing the user's current balance and remaining budget. The system ensures a straightforward workflow that helps users easily navigate through different sections of the app. displays the final summary screen, providing a clear and comprehensive overview of the user's financial status, including balance, remaining budget, and savings progress. highlights the recommendation screen, offering insights into how the user can adjust their spending habits. showcases the complete process of adding income, expenses, and viewing financial insights, ensuring a seamless user experience.

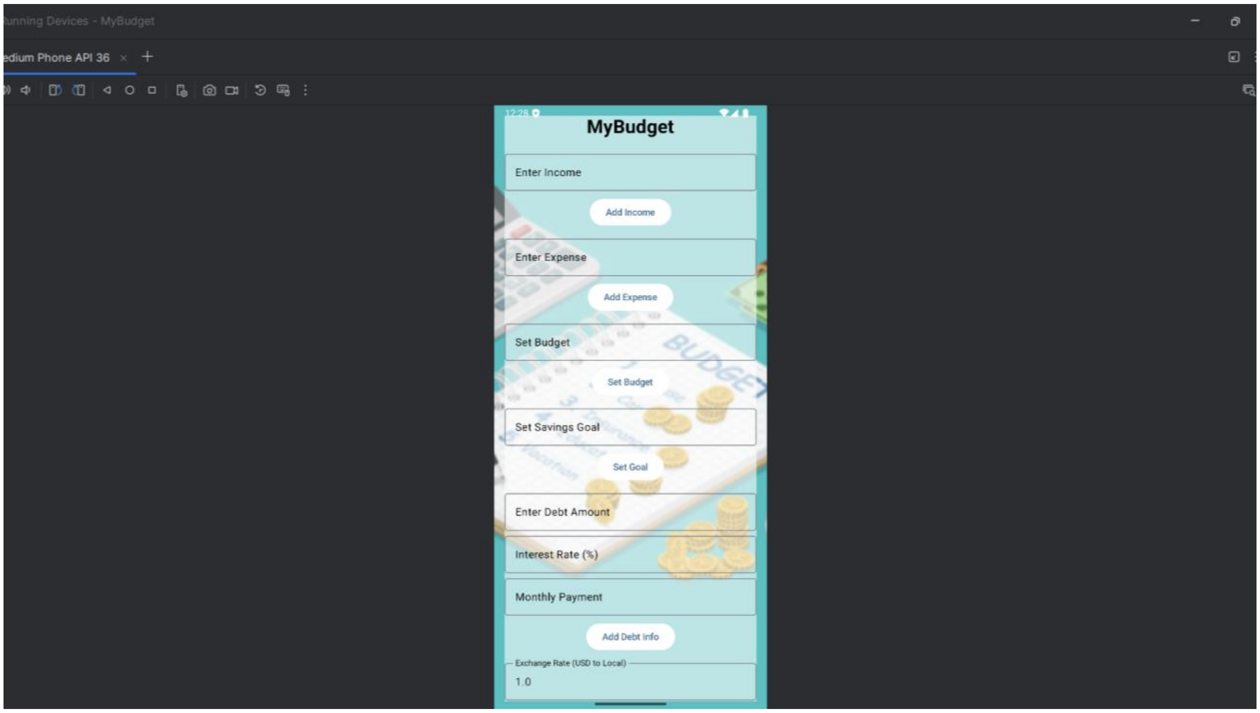



MyBudget
Version control
Medium Phone API 36
MainActivity (1)
Project
MainActivity.kt
// ---------- Budget Screen ----------
@Composable
fun BudgetScreen(onNavigateToNotes: () -> Unit) {
var income by remember { mutableDoubleStateOf(0.0) }
var expense by remember { mutableDoubleStateOf(0.0) }
var budgetLimit by remember { mutableDoubleStateOf(0.0) }
var savingsGoal by remember { mutableDoubleStateOf(0.0) }
var debtAmount by remember { mutableDoubleStateOf(0.0) }
var interestRate by remember { mutableDoubleStateOf(0.0) }
var monthlyPayment by remember { mutableDoubleStateOf(0.0) }
var estimatedTax by remember { mutableDoubleStateOf(0.0) }
var currencyRate by remember { mutableDoubleStateOf(1.0) }
var convertedBalance by remember { mutableDoubleStateOf(0.0) }
var incomeInput by remember { mutableStateOf("") }
var expenseInput by remember { mutableStateOf("") }
var budgetInput by remember { mutableStateOf("") }
var savingsGoalInput by remember { mutableStateOf("") }
var debtInput by remember { mutableStateOf("") }
var interestInput by remember { mutableStateOf("") }
var paymentInput by remember { mutableStateOf("") }
val recurringIncome = 5000.0
val recurringExpense = 1500.0
val forecastedExpense = Random.nextDouble(800.0, 1200.0)
Problems
File
Project Errors
Compose
No problems in MainActivity.kt


MyBudget
Version control
Medium Phone API 36
MainActivity (1)
Project
MainActivity.kt
fun BudgetScreen(onNavigateToNotes: () -> Unit) {
Box(modifier = Modifier.fillMaxSize()) {
) {
OutlinedTextField(value = currencyRate.toString(), onValueChange = { currencyRate = it.toDoubleOrNull() ?: 1.0 }, label = {
Spacer(modifier = Modifier.height(16.dp))
Text("Balance: $${String.format(Locale.US, "%.2f", balance)}", color = Color.Black)
Text("Converted Balance: $${String.format(Locale.US, "%.2f", convertedBalance)}", color = Color.Black)
Text("Remaining Budget: $${String.format(Locale.US, "%.2f", remainingBudget)}", color = Color.Black)
Text("Saved Amount: $${String.format(Locale.US, "%.2f", savedAmount)}", color = Color.Black)
Text("Savings Goal Remaining: $${String.format(Locale.US, "%.2f", savingsRemaining)}", color = Color.Black)
Text("Forecasted Expense (Next Month): $${String.format(Locale.US, "%.2f", forecastedExpense)}", color = Color.Black)
Text("Estimated Tax: $${String.format(Locale.US, "%.2f", estimatedTax)}", color = Color.Black)
Text("Total Debt (with Interest): $${String.format(Locale.US, "%.2f", totalDebt)}", color = Color.Black)
Button(onClick = onNavigateToNotes, modifier = Modifier.padding(top = 16.dp)) {
Text("Go to Notes")
}
if (remainingBudget <= 0) {
Toast.makeText(context, "You exceeded your budget!", Toast.LENGTH_LONG).show()
} else if (remainingBudget <= 50) {
Toast.makeText(context, "You're near your budget limit!", Toast.LENGTH_SHORT).show()
}
Problems
File
Project Errors
Compose
No problems in MainActivity.kt


Project
MainActivity.kt
MyBudget2 [MyBudget]
External Libraries
< Android API 35, extension level
< jbr-21 > C:\Program Files\Andr
aapt2-proto-8.9.2-12782657.jar
aaptcompiler-8.9.2.jar
annotations-4.1.1.4.jar
annotations-13.0.jar
annotations-31.9.2.jar
apksig-8.9.2.jar
apkzlib-8.9.2.jar
fun BudgetScreen(onNavigateToNotes: () -> Unit) {
Box(modifier = Modifier.fillMaxSize()) {
OutlinedTextField(value = budgetInput, onValueChange = { budgetInput = it }, label = { Text("Set Budget", color = Color.Bla
Button(onClick = {
budgetLimit = budgetInput.toDoubleOrNull() ?: 0.0
budgetInput = ""
}, colors = ButtonDefaults.buttonColors(containerColor = Color.White), modifier = Modifier.padding(vertical = 8.dp)) {
Text("Set Budget", color = Color(0xFF406B8C))
// Savings Goal
OutlinedTextField(value = savingsGoalInput, onValueChange = { savingsGoalInput = it }, label = { Text("Set Savings Goal", c
savingsGoal = savingsGoalInput.toDoubleOrNull() ?: 0.0
savingsGoalInput = ""
Text("Set Goal", color = Color(0xFF406B8C))
// Debt
OutlinedTextField(value = debtInput, onValueChange = { debtInput = it }, label = { Text("Enter Debt Amount", color = Color.
OutlinedTextField(value = interestInput, onValueChange = { interestInput = it }, label = { Text("Interest Rate (%)", color
OutlinedTextField(value = paymentInput, onValueChange = { paymentInput = it }, label = { Text("Monthly Payment", color = Co
Problems
File
Project Errors
Compose
No problems in MainActivity.kt


Running Devices - MyBudget
Medium Phone API 36
12:27
MyBudget
Enter Name
Enter PIN
Login


Running Devices - MyBudget
Medium Phone API 36
MyBudget
Enter Income
Add Income
Enter Expense
Add Expense
Set Budget
Set Budget
Set Savings Goal
Set Goal
Enter Debt Amount
Interest Rate (%)
Monthly Payment
Add Debt Info
Exchange Rate (USD to Local)
1.0

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1CONCLUSION

The developed system presents a secure and intelligent solution for detecting fake social media profiles by integrating machine learning with blockchain technology. Utilizing algorithms like Gradient Boosting, Random Forest, and Support Vector Machine (SVM), the system analyzes key profile attributes to identify suspicious behavior with high accuracy. The incorporation of blockchain ensures tamper-proof recording of verification outcomes, fostering trust and transparency in the identity verification process.

The Flask-based web interface provides an intuitive platform for users to submit profile data and receive instant feedback, while the centralized backend processes data efficiently and logs results securely. The system contributes significantly to digital safety by combating the proliferation of fake identities, reinforcing social media integrity, and providing social media administrators with actionable insights. By combining predictive analytics with immutable ledger technology, the project offers a scalable and reliable framework to tackle identity fraud in online environments.

6.2 FUTURE ENHANCEMENT

Future enhancements for this study could include implementing persistent data storage using Room or DataStore to retain income, expenses, and budget records across sessions. Integrating cloud-based user authentication and sync using Firebase can enable multi-device access and secure backup. Visual insights through interactive charts for income and expense trends could improve financial awareness. Smart notifications for budget limits, savings goals, and debt repayment reminders may enhance user engagement. Incorporating currency conversion APIs would allow real-time foreign exchange updates for accurate converted balances. Furthermore, the addition of voice input and multilingual support can make the app more accessible, while offline mode support would improve usability in low-connectivity regions.

REFERENCES

[1] Sharma, R., & Raj, R. "AI-Driven Personal Finance Management Systems: A Review." *International Journal of Computer Applications* 184, no. 32 (2022): 25-30.

[2] Singh, A., & Patel, V. "Integration of Machine Learning in Mobile Finance Applications." In *2023 International Conference on Computational Intelligence and Communication Technology (CICT)*, pp. 102-107. IEEE, 2023.

[3] Kaur, G., & Kumar, N. "Blockchain and Financial Applications: Enhancing Trust and Transparency." *Journal of Financial Innovation and Technology*, vol. 5, no. 1 (2024): 45–56.

[4] Deshmukh, P., & Rao, N. "Securing Digital Wallet Transactions Using Blockchain and Biometric Authentication." In *Proceedings of the 2023 International Conference on Information Security & Privacy*, pp. 86-92. IEEE, 2023.

[5] Reddy, S., & Gupta, A. "Personal Budgeting Applications: A Comparative Study and User-Centric Analysis." *International Journal of Computer and Information Engineering* 17, no. 4 (2023): 205–211.

[6] Banerjee, T., & Das, P. "Smartphone-based Expense Tracker with Voice and NLP Capabilities." In *2024 11th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, pp. 134-139. IEEE, 2024.

[7] Yaga, Dylan, Peter Mell, Nik Roby, and Karen Scarfone. "Blockchain technology overview." *arXiv preprint* arXiv:1906.11078 (2019).

[8] Kapoor, M., & Jain, S. "Gamification in FinTech: Enhancing Engagement in Budgeting Applications." *Journal of Human-Computer Interaction and FinTech* 6, no. 2 (2024): 112–119.

[9] Choudhury, S., & Malik, F. "AI for Tax Estimation and Financial Goal Setting in Mobile Apps." *IEEE Access*, vol. 11 (2023): 55932–55941.

[10] Roy, D., & Srivastava, P. "Privacy-Preserving Techniques for Financial Applications: An Overview." In *Proceedings of the 2024 International Conference on Secure Computing and Communications*, pp. 145-150. Springer, 2024.