



## **FINAL YEAR PROJECT REPORT**

# **TIJORI: BUDGET MONITORING APP**

**In fulfillment of the requirement  
For degree of  
BS (COMPUTER SCIENCES)**

**By**

|                             |                     |
|-----------------------------|---------------------|
| <b>RABIA ARSHAD</b>         | <b>78995 (BSCS)</b> |
| <b>MIRHA HUSSAIN KAMANI</b> | <b>79732 (BSCS)</b> |
| <b>MOAZ RAZA</b>            | <b>79004 (BSCS)</b> |

**SUPERVISED**

**BY  
MS. AZEEMA SADIA**

**BAHRIA UNIVERSITY (KARACHI CAMPUS)**

**SPRING-2025**

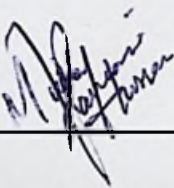
**DECLARATION**

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Signature: 

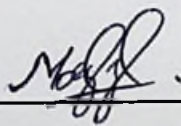
Name: Rabia Arshad

Reg No.: 02-134212-034

Signature: 

Name: Mirha Hussain Kamani

Reg No.: 02-134212-059

Signature: 

Name: Moaz Raza

Reg No.: 02-134212-057

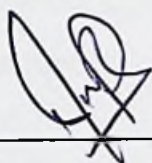
Date: May 5<sup>th</sup> 2025

**APPROVAL FOR SUBMISSION**

We certify that this project report entitled **“TIJORI: BUDGET MONITORING APP”** was prepared by **MIRHA HUSSAIN KAMANI, RABIA ARSHAD, and MOAZ RAZA** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science in **Computer Science** at Bahria University, Karachi.

Approved by,

Signature: \_\_\_\_\_



Supervisor: Miss Azeema Sadia

Date: \_\_\_\_\_

26-May-25

## THREAT RISK MONITORING APP

## ABSTRACT

The copyright of this report belongs to Bahria University according to the Intellectual Property Policy of Bahria University BUORIC-P15 amended on April 2019. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2025 Bahria University. All right reserved.

## **TIJORI: BUDGET MONITORING APP**

### **ABSTRACT**

Tijori is an app that helps users in Pakistan track their finances, understand their investments and improve their financial decisions. This app helps with frequent problems such as not knowing much about finance, not having budgeting tools for everyone and learning little about gold and silver investments. Using Android as its frontend and Java for the backend with Firebase and AWS, Tijori offers competent financial advice each month thanks to Amazon SageMaker and Hugging Face's LLaMA 2 model. To increase the relevance of recommendations, the AI engine was trained with a South Asian financial dataset. Some of its main features are tracking expenses, setting and tracking goals, getting live updates on metal prices, and a chatbot that assists and encourages good financial habits. Raising financial awareness among Pakistanis is the app's primary goal, which the chatbot achieves by providing contextual advice in many languages, including Urdu. The app is built modularly, offering secure login, a user-friendly dashboard and a sequence aimed at helping new users achieve their goals. The thorough testing results revealed accurate recommendations (0.84), responsive actions (210ms) and reliable operation on all devices. The company plans to add banking services and train the AI model with more data as the number of users increases.

## TABLE OF CONTENTS

|                                                           |             |
|-----------------------------------------------------------|-------------|
| <b>DECLARATION</b>                                        | <b>II</b>   |
| <b>APPROVAL FOR SUBMISSION</b>                            | <b>III</b>  |
| <b>ABSTRACT</b>                                           | <b>V</b>    |
| <b>TABLE OF CONTENTS</b>                                  | <b>VI</b>   |
| <b>LIST OF TABLES</b>                                     | <b>XII</b>  |
| <b>LIST OF FIGURES</b>                                    | <b>XIII</b> |
| <b>LIST OF ABBREVIATIONS</b>                              | <b>XIV</b>  |
| <br><b>CHAPTERS</b>                                       |             |
| <b>1. INTRODUCTION</b>                                    | <b>1</b>    |
| 1.1 Background of Personal Finance & Budgeting Apps       | 1           |
| 1.1.1 Rise of FinTech in Daily Life                       | 1           |
| 1.1.2 Global Trends in Budgeting Tools                    | 1           |
| 1.1.3 Budgeting Needs in Emerging Economies like Pakistan | 1           |
| 1.2 Problem Statement                                     | 2           |
| 1.2.1 Financial Illiteracy in Pakistan                    | 2           |
| 1.2.2 Lack of Personalised Budgeting Tools                | 2           |
| 1.2.3 Absence of Local Investment Visibility              | 2           |
| 1.3 Aims and Objectives                                   | 2           |
| 1.3.1 General Aim                                         | 2           |
| 1.3.2 Specific Objectives                                 | 3           |
| 1.3.3 Smart Goals for Tijori                              | 3           |
| 1.4 Scope of Project                                      | 3           |
| 1.4.1 Functional Scope                                    | 3           |

|    |                                                     |    |
|----|-----------------------------------------------------|----|
|    | 1.4.2 Technological Scope (Python, Android, AWS)    | 4  |
|    | 1.4.3 Geographical and User Scope                   | 4  |
| 2. | <b>LITERATURE REVIEW</b>                            | 5  |
|    | 2.1 History and Evolution of Budgeting Apps         | 5  |
|    | 2.1.1 Manual to Digital Transition                  | 5  |
|    | 2.1.2 Milestones in Personal Finance Apps           | 5  |
|    | 2.1.3 Popular App Models Globally                   | 5  |
|    | 2.2 Financial Literacy & Budgeting in South Asia    | 6  |
|    | 2.2.1 Literacy Rates and Mobile Access              | 6  |
|    | 2.2.2 Government and NGO Initiatives                | 6  |
|    | 2.2.3 Role of Mobile Apps in Financial Education    | 6  |
|    | 2.3 Recommendation Systems in FinTech               | 7  |
|    | 2.3.1 Content-Based Filtering(CBF)                  | 7  |
|    | 2.3.2 Collaborative Filtering in Finance            | 7  |
|    | 2.3.3 Hybrid AI-Based Suggestions                   | 7  |
|    | 2.4 UX/UI Patterns in Finance Apps                  | 8  |
|    | 2.4.1 Importance of Minimalist Design               | 8  |
|    | 2.4.2 Colour Theory and Accessibility               | 8  |
|    | 2.4.3 User Onboarding Strategies                    | 9  |
|    | 2.5 Existing Budget Apps vs. Tijori                 | 9  |
|    | 2.5.1 Analysis of Existing Apps                     | 9  |
|    | 2.5.2 Gaps Addressed by Tijori                      | 9  |
|    | 2.5.3 Unique Value Proposition                      | 10 |
| 3. | <b>REQUIREMENT SPECIFICATIONS</b>                   | 11 |
|    | 3.1 Existing System (manual tracking, generic apps) | 11 |
|    | 3.1.1 Manual Budgeting Problems                     | 11 |
|    | 3.1.2 Current App Limitations                       | 11 |

|                                                                   |           |
|-------------------------------------------------------------------|-----------|
| 3.2 Proposed System                                               | 12        |
| 3.2.1 Features of Tijori                                          | 12        |
| 3.2.2 Modules Overview                                            | 14        |
| 3.3 Functional Requirements                                       | 14        |
| 3.3.1 User Management                                             | 14        |
| 3.3.2 Budget Entry and Categorization                             | 15        |
| 3.3.3 Goal Setting and Tracking                                   | 15        |
| 3.3.4 AI Suggestions                                              | 16        |
| 3.4 Non-functional Requirements (security, real-time updates, UX) | 16        |
| 3.4.1 Security & Privacy                                          | 16        |
| 3.4.2 Performance & Availability                                  | 17        |
| 3.4.3 Scalability & Maintainability                               | 17        |
| 3.5 Use Cases                                                     | 17        |
| 3.5.1 Login                                                       | 18        |
| 3.5.2 Track Expenses                                              | 18        |
| 3.5.3 View Suggestions                                            | 18        |
| 3.5.4 Investment Tracking                                         | 18        |
| <b>4. DESIGN AND METHODOLOGY</b>                                  | <b>20</b> |
| 4.1 System Architecture                                           | 20        |
| 4.1.1 Android Frontend                                            | 20        |
| 4.1.2 Python Backend + APIs                                       | 21        |
| 4.1.3 AWS Integration                                             | 21        |
| 4.2 Database Design                                               | 23        |
| 4.2.1 User Profiles                                               | 24        |
| 4.2.2 Expense Tables                                              | 24        |
| 4.2.3 Recommendation Logs                                         | 25        |
| 4.3 Recommendation System Flow                                    | 26        |

|                                                                   |           |
|-------------------------------------------------------------------|-----------|
| 4.3.1 Monthly Suggestions Logic                                   | 27        |
| 4.3.2 Integration with SageMaker and Hugging Face                 | 28        |
| 4.3.3 Personalization Parameters                                  | 28        |
| 4.4 API Integration (Gold rate APIs, metal price graphs)          | 29        |
| 4.4.1 Gold and Silver Rates                                       | 29        |
| 4.4.2 Firebase and AWS Auth APIs                                  | 30        |
| 4.5 UI/UX Planning (Onboarding activities, minimalist dashboard)  | 32        |
| 4.5.1 Onboarding Screens                                          | 32        |
| 4.5.2 Dashboard Layout                                            | 33        |
| 4.5.3 Accessibility Considerations                                | 34        |
| 4.6 Tools Used (Android Studio, Python, AWS, Firebase, REST APIs) | 34        |
| 4.6.1 Android Studio                                              | 34        |
| 4.6.2 Python + FastAPI/Flask                                      | 35        |
| 4.6.3 AWS, Firebase                                               | 35        |
| <b>5. SYSTEM IMPLEMENTATION</b>                                   | <b>36</b> |
| 5.1 Login and Authentication                                      | 36        |
| 5.1.1 Firebase Login Flow                                         | 36        |
| 5.1.2 Forgot Password Logic                                       | 37        |
| 5.2 Expense Tracking Module                                       | 37        |
| 5.2.1 UI Structure                                                | 37        |
| 5.2.2 Data Entry & Validation                                     | 38        |
| 5.2.3 Budget Categories                                           | 38        |
| 5.3 Investment & Market Monitoring                                | 39        |
| 5.3.1 Metal Price API Setup                                       | 39        |
| 5.3.2 Graphing Libraries Used                                     | 40        |
| 5.3.3 Display & Currency Options                                  | 40        |
| 5.4 Financial Preferences & Personalization                       | 40        |

|                                                             |           |
|-------------------------------------------------------------|-----------|
| 5.4.1 Goals Setup                                           | 41        |
| 5.4.2 Recommendation Tuning                                 | 41        |
| 5.5 AI Suggestion Engine                                    | 42        |
| 5.5.1 FinAssist Module                                      | 42        |
| 5.5.2 Text Analysis & Suggestions                           | 43        |
| 5.5.3 Output Display                                        | 44        |
| 5.6 Dashboard Overview                                      | 44        |
| 5.6.1 Aggregated View                                       | 44        |
| 5.6.2 Shortcuts & Stats                                     | 45        |
| 5.7 Onboarding                                              | 45        |
| 5.7.1 Activity Sequence                                     | 45        |
| 5.7.2 Educational Prompts                                   | 45        |
| 5.8 Backend and Cloud Storage                               | 45        |
| 5.8.1 Java Logic                                            | 45        |
| 5.8.2 AWS Server/Storage Details                            | 46        |
| <b>6. TESTING AND EVALUATION</b>                            | <b>47</b> |
| 6.1 Unit Testing                                            | 47        |
| 6.1.1 Activity-wise Testing                                 | 47        |
| 6.1.2 Test Cases for Validation                             | 48        |
| 6.2 Integration Testing                                     | 48        |
| 6.2.1 App + API Testing                                     | 49        |
| 6.2.2 Error Handling                                        | 49        |
| 6.3 User Testing                                            | 50        |
| 6.3.1 Testing with Non-Tech Users                           | 50        |
| 6.3.2 Feedback Collection                                   | 51        |
| 6.4 Performance Evaluation (API latency, app crash reports) | 51        |
| 6.4.1 API Response Time                                     | 51        |

|                                                         |    |
|---------------------------------------------------------|----|
|                                                         | XI |
| 6.4.2 Stability, Battery & Memory Usage                 | 52 |
| 6.5 Limitations                                         | 52 |
| 6.5.1 Offline Constraints                               | 52 |
| 6.5.2 Lack of Urdu Support                              | 53 |
| 6.5.3 Bank Linking Unavailable                          | 53 |
| 7. <b>CONCLUSION AND FUTURE WORK</b>                    | 54 |
| 7.1 Summary of Project Outcomes                         | 54 |
| 7.1.1 What Was Achieved                                 | 54 |
| 7.1.2 How It Solves the Problem                         | 54 |
| 7.2 Challenges Faced (time, training AI, API stability) | 55 |
| 7.2.1 Technical                                         | 55 |
| 7.2.2 Resource                                          | 55 |
| 7.2.3 Time Management                                   | 56 |
| 7.3 Future Enhancements                                 | 56 |
| 7.3.1 Bank Linking                                      | 56 |
| 7.3.2 Urdu/Multilingual Support                         | 57 |
| 7.3.3 Advanced Budget Prediction                        | 57 |
| 7.4 Final Remarks                                       | 57 |
| 7.4.1 Impact of Tijori                                  | 57 |
| 7.4.2 Personal Learning Outcomes                        | 58 |
| <b>REFERENCES</b>                                       | 59 |
| <b>APPENDICES</b>                                       | 67 |