**ABSTRACT**

In today's digital world, managing personal finances has become more important and challenging than ever. People often struggle to keep track of their income, expenses, and savings due to a lack of time or proper tools. The **Powered Budget Management System** is developed to solve these problems by providing a smart, easy-to-use, and efficient solution for budget tracking.

This web-based application allows users to create multiple budgets, add income and expense records, and monitor their financial habits using interactive charts and dashboards. One of the key features of this system is the integration of Artificial Intelligence using the **Google Gemini API**, which enables users to scan receipts and automatically extract transaction details. This helps reduce manual entry and improves accuracy.

The system is built using modern technologies like **React.js** for the frontend, **Node.js** and **Express.js** for the backend, **MySQL** for data storage, and **Prisma ORM** for easy and type-safe database interactions. It also includes secure login using JWT authentication and a responsive design that works on all devices.

This project aims to make financial management simple, secure, and intelligent for students and working individuals, helping them make better spending decisions and achieve their financial goals.

**Keywords**: Budget Tracker, AI Receipt Scanner, React.js, Node.js, Prisma ORM, MySQL, Financial Dashboard, JWT Authentication

**TABLE OF CONTENTS**

**ABSTRACT**  
**LIST OF TABLES**  
**LIST OF FIGURES**  
**LIST OF ABBREVIATIONS AND SYMBOLS**

**1. INTRODUCTION 1**  
  1.1 Overview 1  
  1.2 Problem Statement 1  
  1.3 Objectives 1  
  1.4 Motivation 1  
  1.5 Hardware and Software Requirements Specification Document 2  
  1.6 Project Budget Plan 2

**2. LITERATURE SURVEY 3**

**3. PROBLEM ANALYSIS & DESIGN 4**  
  3.1 Existing System 3  
  3.2 Proposed System 4  
  3.3 Identified Tools / Libraries / Software 4  
  3.4 Architectural Block Diagram & Corresponding System Modeling 5

**4. IMPLEMENTATION 9**  
  4.1 Overview of System Implementation 9  
  4.2 Module Description 9  
  4.3 Code Snippets 10

**5. TESTING 11**  
  5.1 Test Design with Testcases 11  
  5.2 Test Report 18

**6. RESULTS 19**  
  6.1 Results Snippets 19

**7. CONCLUSION AND FUTURE SCOPE 20**

**REFERENCES 21**

**PLAGARISM REPORT 22**

**LIST OF TABLES**

| **Table No.** | **Caption** | **Page No.** |
| --- | --- | --- |
| Table 1 | Project Budget Plan | 2 |
| Table 2 | Literature Survey Table | 3 |
| Table 3 | blackbox testing table | 14 |
| Table 4 | white box testing table | 15 |
| Table 5 | Integration testing table | 18 |
| Table 6 | Test Report | 18 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **LIST OF FIGURES**   | **Figure No.** | **Caption** | **Page No.** | | --- | --- | --- | | Figure 1 | Architectural Block Diagram | 5 | | Figure 2 | Data Flow Diagram | 5 | | Figure 3 | Use Case Diagram | 6 | | Figure 4 | ER Diagram | 7 | | Figure 5 | React.js Sample Code Snippet | 10 | | Figure 6 | Node.js Sample Code Snippet | 10 | | Figure 7 | Authentication code snippet for testing | 13 | | Figure 8 | Budgets code snippet for testing | 13 | | Figure 9 | Transaction code snippet for testing | 13 | | Figure 10 | Budgets Output | 19 | | Figure 11 | AllTransactions Output | 19 | | Figure 12 | Dashboard Output | 19 | |
|  |

**LIST OF ABBREVATIONS**

| **Sl. No.** | **Abbreviation** | **Full Form** |
| --- | --- | --- |
| 1 | AI | Artificial Intelligence |
| 2 | UI | User Interface |
| 3 | JWT | JSON Web Token |
| 4 | API | Application Programming Interface |
| 5 | ORM | Object Relational Mapping |
| 6 | MERN | MongoDB, Express.js, React.js, Node.js |
| 7 | CRUD | Create, Read, Update, Delete |
| 8 | DB | Database |
| 9 | IDE | Integrated Development Environment |
| 10 | VS Code | Visual Studio Code |
| 11 | CSS | Cascading Style Sheets |