Software Requirements Specification

for

CIBIL Score Tracker

Version 1.0 approved

Prepared by Dharaneesh V

Coimbatore Institute of Technology

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for the CIBIL Score Tracker System (CSTS). This system is designed to provide both end-users and administrators with a comprehensive platform for monitoring, analyzing, and managing credit scores. The product will allow users to fetch their CIBIL score securely, view personalized tips for improvement, set financial goals, track their progress, and manage their credit profile. For administrators, the system will include features for updating credit improvement tips, generating recommendations, managing user accounts, and integrating external APIs for credit data retrieval. This SRS describes the full scope of the system, covering user-facing modules, administrative tools, and backend functionality required for accurate credit score management. The first release of the CSTS will focus on the core functionalities depicted in the use case diagram, including score retrieval, recommendation generation, and user progress tracking.

1.2 Document Conventions

This Software Requirements Specification (SRS) adheres to IEEE guidelines for SRS documentation (IEEE 830-1998). Each section and subsection is numbered hierarchically (e.g., 1.0, 1.1, 1.2) to ensure clarity and easy reference. UML (Unified Modeling Language) notations are used for diagrams, including the use case diagram attached, to illustrate the interaction between actors and the system. Functional requirements are expressed in plain text, with emphasis given through bold font for system names, italics for references or external documents, and CAPITALIZATION where system actions are critical. Priorities for higher-level requirements are inherited by detailed requirements unless explicitly overridden. This ensures consistency and reduces redundancy in requirement descriptions

1.3 Intended Audience and Reading Suggestions

This document is intended for a wide range of stakeholders involved in the development and deployment of the CIBIL Score Tracker.

Developers will use the requirements to implement functional modules, such as score fetching, dashboard creation, and recommendation logic.

Testers and QA Engineers will refer to this document to validate whether the system meets the specified requirements through functional and performance testing.

Project Managers will use it for planning, resource allocation, scheduling, and monitoring progress.

Administrators will reference the sections related to system management, such as user handling, updating tips, and report generation.

End Users (Customers) will benefit from understanding how the system supports their needs, such as monitoring scores, tracking progress, and managing goals.

The document begins with the introduction and scope, followed by detailed system descriptions, functional requirements, non-functional requirements, and external interface specifications. Readers are advised to first review the introduction and project scope to gain an overview before diving into the detailed requirements.

1.4 Project Scope

The CIBIL Score Tracker System aims to empower individuals by giving them complete visibility into their credit score and providing actionable insights for improvement. Users can register/login, fetch their CIBIL score, and interact with a personalized dashboard that highlights key features such as viewing tips for credit improvement, setting and tracking financial goals, analyzing credit behavior, and reviewing score history. The system provides a secure authentication mechanism and ensures that sensitive financial data is handled responsibly.

From the administrative perspective, the system ensures operational efficiency by providing tools to manage user accounts, view system reports, update financial tips and recommendations, and maintain credit score data through APIs. By combining data analytics with personalized recommendations, the system not only informs users of their current credit health but also guides them toward achieving a stronger credit profile.

The overall business objective of the project is to contribute toward financial literacy, better credit management, and responsible borrowing practices. By aligning with corporate financial goals, the CSTS supports users in making informed financial decisions while enabling administrators to maintain system accuracy and relevance. This system serves as a stepping stone toward broader financial technology solutions that help both individuals and organizations manage credit responsibly.

1.5 References

- IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.
- UML Notation Guide for Use Case Diagrams.
- Official CIBIL documentation and API guidelines: https://www.cibil.com.
- Reserve Bank of India (RBI) Guidelines for Credit Information Companies.
- Project Vision Document for **CIBIL Score Tracker System**, Version 1.0, 2025.
- User Interface Guidelines and Design Standards (internal reference, 2025).

2. Overall Description

2.1 Product Perspective

The **CIBIL Score Tracker System (CSTS)** is a **web-based application** developed using HTML, CSS, JavaScript, PHP, and MySQL. It is a new standalone product that allows users to fetch, track, and improve their credit scores. The system consists of two major components: the **user module** and the **admin module**. The user module enables individuals to register/login, fetch their credit score, set credit goals, and receive personalized recommendations. The admin module provides features for updating financial tips, managing users, and generating reports.

The system follows a **client-server architecture** where the front-end is implemented using HTML, CSS, and JavaScript for user interaction, while the back-end uses PHP to process requests, interact with the database, and communicate with external APIs (like CIBIL's credit score API). MySQL serves as the central database for securely storing user data, score history, and system-generated recommendations.

2.2 Product Features

The main features of the **CIBIL Score Tracker System** include:

- **User Registration & Authentication**: Secure login and signup functionality managed via PHP and MySQL.
- **Personalized Dashboard**: Displays user's latest CIBIL score, goal progress, tips, and recommendations.
- **Fetch CIBIL Score**: Integrates with CIBIL API (or test API) to retrieve the user's latest credit score.
- **Profile & Settings Management**: Users can update personal details and preferences.
- **Credit Improvement Tips**: System and admin-provided suggestions to improve financial health.
- **Credit Goal Setting & Tracking**: Allows users to define credit goals and monitor progress.
- **Score History Tracking**: Stores and displays past CIBIL scores for analysis.
- **Behavior Analysis & Recommendations**: Identifies credit behavior patterns and generates improvement advice.
- **Admin Functions**: Includes user management, updating recommendations, generating system reports, and managing the credit score API connection.

2.3 User Classes and Characteristics

There are two main user classes:

1. **End Users (General Users)**
   - Individuals who want to track and improve their credit scores.

  - Technical expertise: Basic computer and internet knowledge, no programming skills required.
  - Privileges: Limited to personal account functions (dashboard, goals, tips, score history).

2. **Administrators (Admins)**
   - Financial and technical staff responsible for managing the system.
   - Technical expertise: Familiar with credit scoring concepts and comfortable with web applications.
   - Privileges: Manage users, update tips and recommendations, view system reports, and handle API/data integration.

2.4 Operating Environment

The CIBIL Score Tracker is a web-based system that will run in the following environment:

- **Frontend**: HTML5, CSS3, JavaScript (for UI, validation, and interactivity).
- **Backend**: PHP 7.x/8.x (for server-side processing, authentication, and business logic).
- **Database**: MySQL 5.7 or later for storing user data, credit scores, and recommendations.
- **Web Server**: Apache HTTP Server (XAMPP/WAMP/LAMP stack supported).
- **Browsers Supported**: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari (latest versions).
- **Hardware**: Any modern PC or smartphone with 2GB+ RAM and stable internet connection.
- **Operating System**: Cross-platform (Windows, Linux, macOS).

2.5 Design and Implementation Constraints

- **Technology Constraint**: System is restricted to **HTML, CSS, JavaScript, PHP, and MySQL** as per project requirements.
- **Security Constraint**: User passwords must be hashed (e.g., SHA-256 or bcrypt) in MySQL database. Sensitive financial data must be encrypted during storage and transmission.
- **API Constraint**: Fetching CIBIL scores depends on availability of external API (or demo/test API).
- **Performance Constraint**: The system should fetch and display credit scores within 3 seconds on a stable internet connection.

- **Browser Compatibility**: Must work consistently across major browsers.
- **Database Constraint**: Relational schema with normalization to avoid redundancy.
- **Maintenance Constraint**: Admin must regularly update financial tips and recommendations stored in MySQL database.

2.6 User Documentation

The following documentation will be delivered with the system:

- **User Guide (HTML/PDF)** – Explains how users can register, log in, view scores, and set goals.
- **Admin Guide** – Explains how admins can manage users, update tips, and generate reports.
- **Online Help Section** – FAQs and help topics integrated into the dashboard.
- **Tutorials** – Step-by-step instructions and screenshots to guide first-time users.

2.7 Assumptions and Dependencies

- Users must have a stable internet connection to fetch live CIBIL scores.
- The system assumes that the **CIBIL API or test API** will be available and reliable.
- The system depends on PHP and MySQL support on the web server.
- It assumes that users provide valid and accurate information during registration.
- Browser support is limited to modern browsers; outdated browsers may not render the system correctly.
- Data security assumes HTTPS implementation on the deployment server.
- Any change in external API policies may require modification of the system.

3. System Features

3.1 User Registration and Login

3.1.1 Description and Priority

This feature allows users to create an account and log in securely. It ensures only authenticated users access personalized data.
Priority: High (Benefit = 9, Penalty = 8, Cost = 4, Risk = 5).

3.1.2 Stimulus/Response Sequences

- **Stimulus:** User clicks "Register" → enters name, email, password → submits form.
- **Response:** System validates inputs, hashes password, stores data in MySQL, and confirms registration.
- **Stimulus:** User enters email and password on login form.
- **Response:** System checks credentials, starts session, redirects to dashboard.

3.1.3 Functional Requirements

- **REQ-1.1:** The system shall validate all inputs (unique email, strong password).
- **REQ-1.2:** The system shall store passwords in encrypted format.
- **REQ-1.3:** The system shall authenticate users and maintain session state.
- **REQ-1.4:** The system shall reject invalid login attempts and show error messages.
- **REQ-1.5:** The system shall log out users and destroy session when requested.

3.2 Dashboard

3.2.1 Description and Priority

The dashboard is the main user interface where CIBIL score, score history, credit goals, and tips are displayed.
Priority: High (Benefit = 9, Penalty = 7, Cost = 5, Risk = 4).

3.2.2 3.2.2 Stimulus/Response Sequences

- **Stimulus:** User logs in and opens dashboard.
- **Response:** System retrieves data (latest score, tips, goals, history) from MySQL and displays it dynamically.

3.2.3 3.2.3 Functional Requirements

- **REQ-2.1:** The system shall display the latest CIBIL score.
- **REQ-2.2:** The system shall present user-specific credit improvement tips.
- **REQ-2.3:** The system shall provide a score history graph.
- **REQ-2.4:** The system shall allow quick navigation to goals and recommendations.

3.3 Fetch CIBIL Score

3.3.1 Description and Priority

This feature fetches the user’s updated CIBIL score via an API or a simulated backend function. **Priority:** High (Benefit = 9, Penalty = 9, Cost = 6, Risk = 6).

3.3.2 Stimulus/Response Sequences

- **Stimulus:** User clicks “Fetch Score”.
- **Response:** System queries API → retrieves score → updates MySQL → refreshes dashboard.

3.3.3 Functional Requirements

- **REQ-3.1:** The system shall connect securely to the CIBIL API.
- **REQ-3.2:** The system shall update the user’s score in the database.
- **REQ-3.3:** The system shall handle API failures and notify the user.
- **REQ-3.4:** The system shall show the latest score in real-time.

3.4 Credit Goals

3.4.1 Description and Priority

This feature lets users set and track personal credit goals (e.g., reduce debt, improve score). **Priority:** Medium (Benefit = 8, Penalty = 6, Cost = 5, Risk = 4).

3.4.2 Stimulus/Response Sequences

- **Stimulus:** User enters a goal (e.g., "Improve score to 750 in 6 months").
- **Response:** System saves it in MySQL and tracks progress against updates.

3.4.3 Functional Requirements

- **REQ-4.1:** The system shall allow users to create, edit, and delete goals.
- **REQ-4.2:** The system shall calculate progress and show status updates.
- **REQ-4.3:** The system shall notify users if progress is off-track.

3.5 Tips and Recommendations

3.5.1 Description and Priority

Provides personalized suggestions for improving credit health, added by admin or system-generated. **Priority:** High (Benefit = 8, Penalty = 7, Cost = 3, Risk = 4).

3.5.2 Stimulus/Response Sequences

- **Stimulus:** User opens "Tips" tab in dashboard.
- **Response:** System fetches tips from database and displays them.

3.5.3 Functional Requirements

- **REQ-5.1:** The system shall display financial tips from the database.
- **REQ-5.2:** The system shall allow admins to manage (add/edit/delete) tips.
- **REQ-5.3:** The system shall categorize tips (High/Medium/Low Impact).

3.6 Admin User Management

3.6.1 Description and Priority

Admin can manage registered users: view, suspend, or delete accounts. **Priority:** High (Benefit = 9, Penalty = 8, Cost = 4, Risk = 5).

3.6.2 Stimulus/Response Sequences

- **Stimulus:** Admin logs in → opens user management panel.
- **Response:** System displays user list with action options.

3.6.3 Functional Requirements

- **REQ-6.1:** The system shall allow admins to view all users.
- **REQ-6.2:** The system shall allow admins to suspend or delete accounts.
- **REQ-6.3:** The system shall restrict non-admins from accessing admin features.
- **REQ-6.4:** The system shall record all admin actions for auditing.

3.7 Reports and Analytics

3.7.1 3.7.1 Description and Priority

Generates reports on user activity, score trends, and credit improvement statistics. **Priority:** Medium (Benefit = 7, Penalty = 5, Cost = 5, Risk = 4).

3.7.2 3.7.2 Stimulus/Response Sequences

- **Stimulus:** Admin clicks “Generate Report”.
- **Response:** System compiles data and presents report in table/graph format.

3.7.3 3.7.3 Functional Requirements

- **REQ-7.1:** The system shall generate reports on average scores, goals achieved, and usage.
- **REQ-7.2:** The system shall allow report export in CSV/PDF.
- **REQ-7.3:** The system shall allow filtering by date range.

4. External Interface Requirements

4.1 User Interfaces

The **CIBIL Score Tracker** will have a **responsive web-based user interface** built using **HTML, CSS, and JavaScript**. The design will follow a clean, minimal style with consistent fonts, color schemes, and layout across all pages. Key components:

- **Login/Registration Screen:** Text boxes for email, password, and validation messages. Buttons for login, registration, and password reset.
- **Dashboard:** Displays the latest CIBIL score, graphical score history (line/bar chart using JS libraries), progress toward goals, and personalized tips.
- **Goals Section:** Form-based interface for users to add, edit, and delete credit goals.
- **Tips Section:** Static and dynamic tips presented in card-style UI, with filtering options.
- **Admin Panel:** Table-based UI showing all registered users, action buttons (suspend/delete), and analytics graphs.
- **Standard Elements:** Navigation bar, help button, error message alerts, and confirmation modals.
- **GUI Standards:** Consistent use of form validation, success/error popups, tooltips, and color codes (e.g., green for good credit score, red for poor score).

4.2 Hardware Interfaces

Since this is a **web-based application**, no specialized hardware is required. The system shall run on:

- **Client Devices:** Desktops, laptops, tablets, and smartphones with modern browsers (Chrome, Edge, Firefox, Safari). Minimum screen resolution supported: **1280x720**.
- **Server:** A standard server with PHP 7.x or later, MySQL 8.0 or later, and Apache/Nginx as the web server.
- **Data Flow:** User input is captured from keyboard/touch devices, transmitted over HTTPS, processed by server hardware, and results displayed back on the user's device.
- **Storage:** Data will be stored in a relational MySQL database hosted on the server. Backup hardware should be available for recovery.

4.3 Software Interfaces

The system will integrate with various **software components** for full functionality:

- **Operating System:** Runs on Windows/Linux server environments.
- **Web Server:** Apache or Nginx to handle HTTP requests.
- **Backend:** PHP (7.x or higher) for server-side scripting and logic.
- **Database:** MySQL 8.0 for storing user credentials, score history, goals, and tips.
- **Frontend:** HTML5, CSS3, JavaScript (with optional libraries like Chart.js for score graphs).
- **APIs:** The system may integrate with third-party financial APIs (such as CIBIL or mock credit score providers) to fetch real-time score updates.
- **Data Exchange:** JSON format will be used for client-server communication (AJAX requests).
- **Shared Data:** User profiles, authentication tokens, and score history records will be securely exchanged between components.

4.4 Communications Interfaces

The CIBIL Score Tracker will rely on **internet-based communication protocols** to function properly:

- **Web Communication:** The application will use HTTP/HTTPS protocols. HTTPS with SSL/TLS encryption will be mandatory to ensure secure communication.
- **Client-Server Data Flow:** User actions (login, fetch score, update goals) will send AJAX requests from the client to the server, which responds with structured JSON data.
- **Email Notifications:** Integration with SMTP services may be used to send password reset links or alerts regarding score updates.
- **Network Requirements:** Minimum bandwidth requirement is 1 Mbps for smooth operation.
- **Security:** Sensitive data such as login credentials and API keys will be transmitted over secure channels using HTTPS. Passwords will never be sent in plain text.
- **Error Handling:** If communication fails (e.g., API not responding), the system will display an error message and log the incident for administrators.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The **CIBIL Score Tracker** must provide a responsive and efficient user experience.

- **System Response Time:** All dashboard actions (login, fetching score, updating goals) shall complete within **2–3 seconds** under normal internet speed (≥1 Mbps).
- **Concurrent Users:** The system shall support at least **200 simultaneous users** without performance degradation.
- **Database Queries:** Queries to the MySQL database (e.g., retrieving score history, tips, or goals) shall execute within **1 second**.
- **API Fetching:** When fetching real-time CIBIL score via third-party API, the system shall handle response times up to **5 seconds** gracefully, with a loading indicator.
- **Scalability:** The application shall be designed to scale horizontally by adding more servers in case of increased demand.

5.2 Safety Requirements

The application does not directly control physical devices but involves sensitive financial information, so safety in terms of **data protection and reliability** is critical.

- **Data Integrity:** The system shall ensure that no score or goal data is lost during server crashes or unexpected shutdowns by using **regular database backups**.
- **Transaction Safety:** All updates to user accounts, goals, and scores must be atomic; partial updates are not permitted.
- **Error Handling:** The system shall provide meaningful error messages without exposing technical details that could compromise security.
- **Disaster Recovery:** Regular backup and restore mechanisms shall be implemented, with weekly full backups and daily incremental backups.

5.3 Security Requirements

Security is crucial, as the system deals with **personal and financial data**.

- **Authentication:** All users shall authenticate using email and password before accessing the system.
- **Password Management:** Passwords shall be stored using **secure hashing algorithms (e.g., bcrypt or SHA-256 with salt)**, never in plain text.
- **Session Management:** Sessions shall expire after **15 minutes of inactivity** to reduce risk of unauthorized access.
- **Data Privacy:** Sensitive user data (CIBIL scores, personal details) shall be encrypted in transmission using **HTTPS with SSL/TLS**.
- **Authorization:** Only admin users shall have access to the admin panel for managing accounts and reports.
- **Protection Against Attacks:** The system shall include safeguards against SQL injection, XSS (Cross-Site Scripting), and brute-force login attempts.
- **Regulatory Compliance:** The system shall align with **data protection regulations (such as GDPR principles)** regarding user consent and data handling.

5.4 Software Quality Attributes

The system shall adhere to the following quality attributes:

- **Usability:** The user interface shall be intuitive, with clear navigation and consistent design for both novice and experienced users.
- **Reliability:** The application shall have an uptime of **at least 99%**, excluding scheduled maintenance.
- **Maintainability:** The codebase shall follow **modular coding practices** in PHP and JavaScript, making it easy to update and debug.
- **Portability:** The application shall run smoothly across major browsers (Chrome, Firefox, Safari, Edge) and operating systems (Windows, macOS, Linux, Android, iOS).
- **Interoperability:** The system shall integrate seamlessly with third-party APIs for CIBIL score fetching and SMTP services for notifications.
- **Adaptability:** The system shall support both desktop and mobile devices with a responsive design.

- **Robustness:** The system shall continue functioning gracefully even under unexpected conditions, such as API failure or server overload.
- **Testability:** Each module (login, dashboard, goals, tips, admin panel) shall be independently testable with unit and integration tests.

6. Other Requirements

6.1 Database Requirements

- The system shall use **MySQL 8.0** as the primary database management system.
- All tables shall follow **normalization up to 3NF** to avoid redundancy and ensure data integrity.
- Primary keys and foreign keys shall be properly defined for relationships (e.g., users → scores → goals).
- Database shall enforce constraints such as **NOT NULL, UNIQUE, and FOREIGN KEY** to maintain consistency.
- Backup mechanisms shall include **daily incremental backups** and **weekly full backups** stored on a secure server.
- Data retention policy: User data shall be retained for **minimum 3 years** unless a deletion request is raised by the user.

6.2 Internationalization and Localization Requirements

- The system shall support **English** as the default language, but it shall be designed to allow easy extension to other languages.
- All date and currency formats shall be **configurable based on user location** (e.g., INR for Indian users, USD for international users).
- The user interface shall use **Unicode (UTF-8 encoding)** to support special characters and multilingual text.

6.3 Legal and Regulatory Requirements

- The system shall comply with **Indian IT Act 2000** and applicable **data protection regulations** for handling sensitive personal data.

- If expanded internationally, the system shall align with **GDPR (General Data Protection Regulation)** for European users.
- The application shall provide a **privacy policy** and **terms of service** for all users at the time of registration.
- User consent shall be explicitly obtained before accessing or storing credit-related data.
- The system shall not share or sell user financial data to unauthorized third parties.

6.4 Reuse and Scalability Requirements

- The system shall be designed in a **modular structure** (separating frontend, backend, and database layers) to facilitate reuse in future credit-related projects.
- The backend API layer shall be reusable for future integrations with mobile applications (Android/iOS).
- The system shall be scalable to handle an increased user base by upgrading server infrastructure or moving to cloud hosting solutions like AWS or Azure.

6.5 Reporting and Audit Requirements

- The system shall maintain an **audit trail** of user activities (e.g., login attempts, score fetches, goal updates).
- Admins shall be able to generate reports of system usage, score trends, and goal completion rates.
- Logs shall be stored for **at least 1 year** for compliance and troubleshooting.

6.6 Usability and Accessibility Requirements

- The application shall follow **WCAG 2.1 accessibility standards**, ensuring it can be used by people with disabilities.
- Font sizes, color contrasts, and interactive elements shall be optimized for readability and usability.
- The system shall support both **desktop and mobile browsers** with a responsive design.

Appendix A: Glossary

- **CIBIL Score**: Credit Information Bureau (India) Limited score; a 3-digit number ranging from 300–900 that indicates a person's creditworthiness.

- **Dashboard**: The main interface that displays a user's CIBIL score, credit history, and financial recommendations.

- **Goal**: A user-defined financial milestone (e.g., improving credit score to 750 in six months).

- **API (Application Programming Interface)**: A software intermediary that allows the application to fetch real-time credit scores from external services.

- **Authentication**: Process of verifying a user's identity through login credentials.

- **Authorization**: Process of granting user roles and permissions (e.g., admin vs. normal user).

- **MySQL**: Relational Database Management System (RDBMS) used to store and manage project data.

- **PHP**: Hypertext Preprocessor, the server-side scripting language used for backend logic.

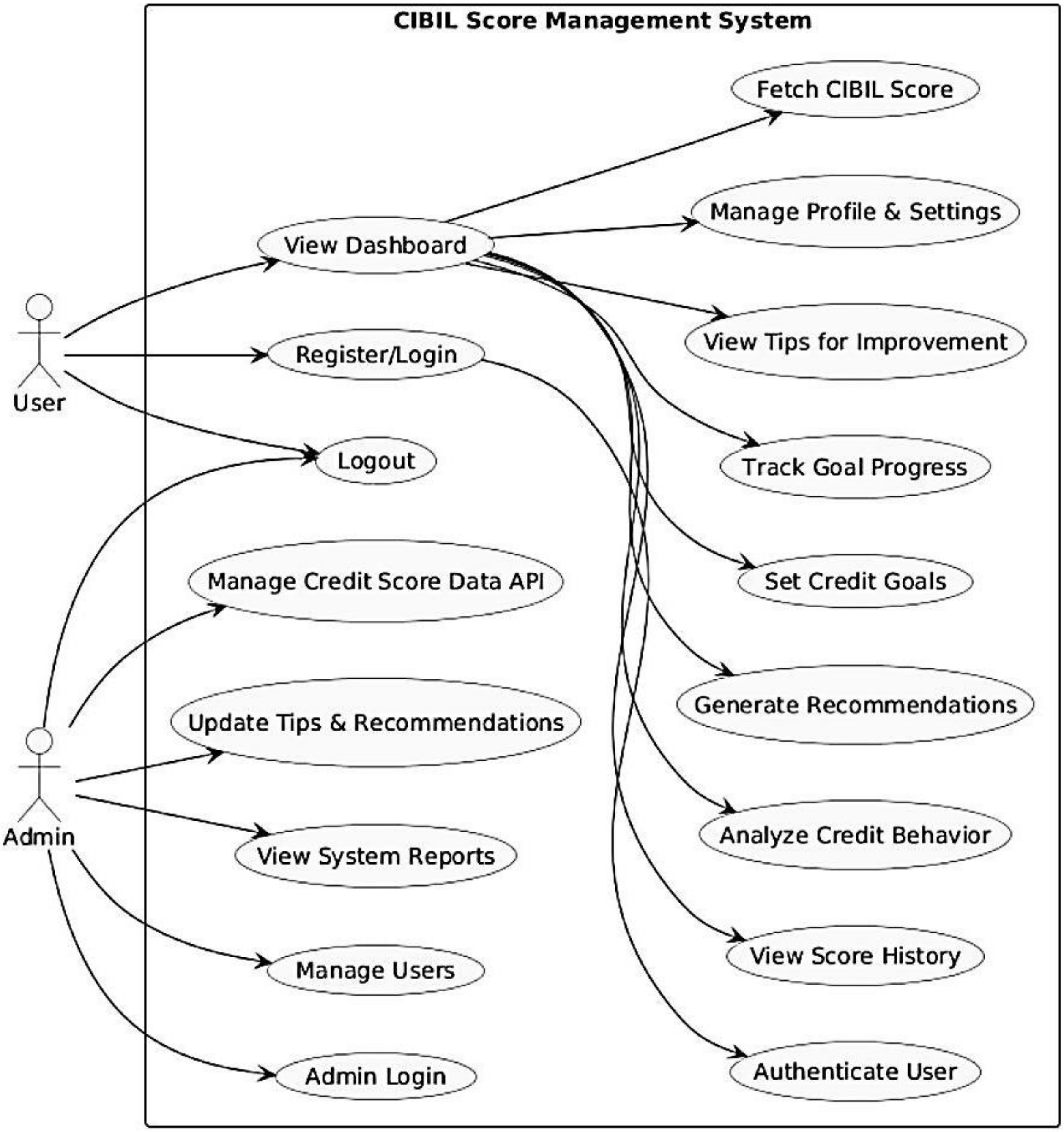
- **Responsive Design**: Web design approach ensuring usability across different devices (desktop, tablet, mobile).

- **Encryption**: Process of converting sensitive information (like passwords) into a secure format to prevent unauthorized access.

- **Audit Trail**: A chronological record of system activities for security and compliance purposes.

Appendix B: Analysis Models

Use case Diagram



Use case Description

UC 1: Register/Login
Primary Actor: User/Admin
Secondary Actor: Authentication Service
Precondition: User/Admin is not logged in
Trigger: User/Admin opens the system

Main Success Scenario:
1. User/Admin provides login credentials
2. System validates credentials
3. User/Admin is redirected to their respective dashboard

Exception Scenarios:
1. Invalid credentials → System shows error message
2. Account locked or inactive → System shows "Contact Admin"

UC 2: Authenticate User
Primary Actor: System
Precondition: User/Admin credentials submitted
Trigger: Login attempt made

Main Success Scenario:
1. System checks credentials
2. If valid, session is created

Exception Scenario:
1. Invalid credentials → Deny access

UC 3: View Dashboard
Primary Actor: User
Precondition: User is authenticated
Trigger: Successful login

Main Success Scenario:
1. System displays dashboard
2. Shows quick stats, score, tips, etc.

UC 4: Fetch CIBIL Score
Primary Actor: User
Secondary Actor: External CIBIL API
Precondition: User is logged in
Trigger: User clicks “Fetch CIBIL Score”

Main Success Scenario:
1. System calls external CIBIL API
2. Receives latest score
3. Displays it on the dashboard
4. Saves it in score history

Exception Scenario:
1. API call fails → Show “Try again later”
2. No PAN linked → Ask user to update profile

UC 5: View Score History
Primary Actor: User
Precondition: Score data available
Trigger: User selects “View History”

Main Success Scenario:
1. System retrieves historical score entries
2. Displays in table/graph format

UC 6: Analyze Credit Behavior
Primary Actor: System
Precondition: Score history available
Trigger: User navigates to “Analysis”

Main Success Scenario:
1. System analyzes past scores, trends
2. Detects delays, usage patterns
3. Generates a summary report

UC 7: Generate Recommendations
Primary Actor: System
Precondition: Analysis is complete
Trigger: System completes behavior analysis

Main Success Scenario:
1. System suggests actions (e.g., lower usage)
2. Recommendations displayed to user

UC 8: Set Credit Goals
Primary Actor: User
Precondition: Score is known
Trigger: User wants to improve credit

Main Success Scenario:
1. User defines target score
2. System saves goal with timeframe

UC 9: Track Goal Progress
Primary Actor: User
Precondition: Goals are set
Trigger: User opens goal tracker

Main Success Scenario:
1. System compares current score with goal
2. Displays progress %

UC 10: View Tips for Improvement
Primary Actor: User
Precondition: User is logged in
Trigger: User navigates to tips section

Main Success Scenario:
1. System fetches improvement tips
2. Displays relevant suggestions

UC 11: Manage Profile & Settings
Primary Actor: User
Precondition: User is logged in
Trigger: User edits profile

Main Success Scenario:
1. User updates profile details
2. System saves changes

UC 12: Admin Login
Primary Actor: Admin
Precondition: Admin is not logged in
Trigger: Admin attempts login

Main Success Scenario:
1. Admin provides credentials
2. System authenticates and grants access

UC 13: Manage Credit Score Data API
Primary Actor: Admin
Precondition: Admin is logged in
Trigger: Admin opens API settings

Main Success Scenario:
1. Admin updates API keys or endpoints
2. System saves configuration

UC 14: Update Tips & Recommendations
Primary Actor: Admin
Precondition: Admin is logged in
Trigger: Admin accesses tips manager

Main Success Scenario:
1. Admin adds/edits tips
2. Changes are reflected for users

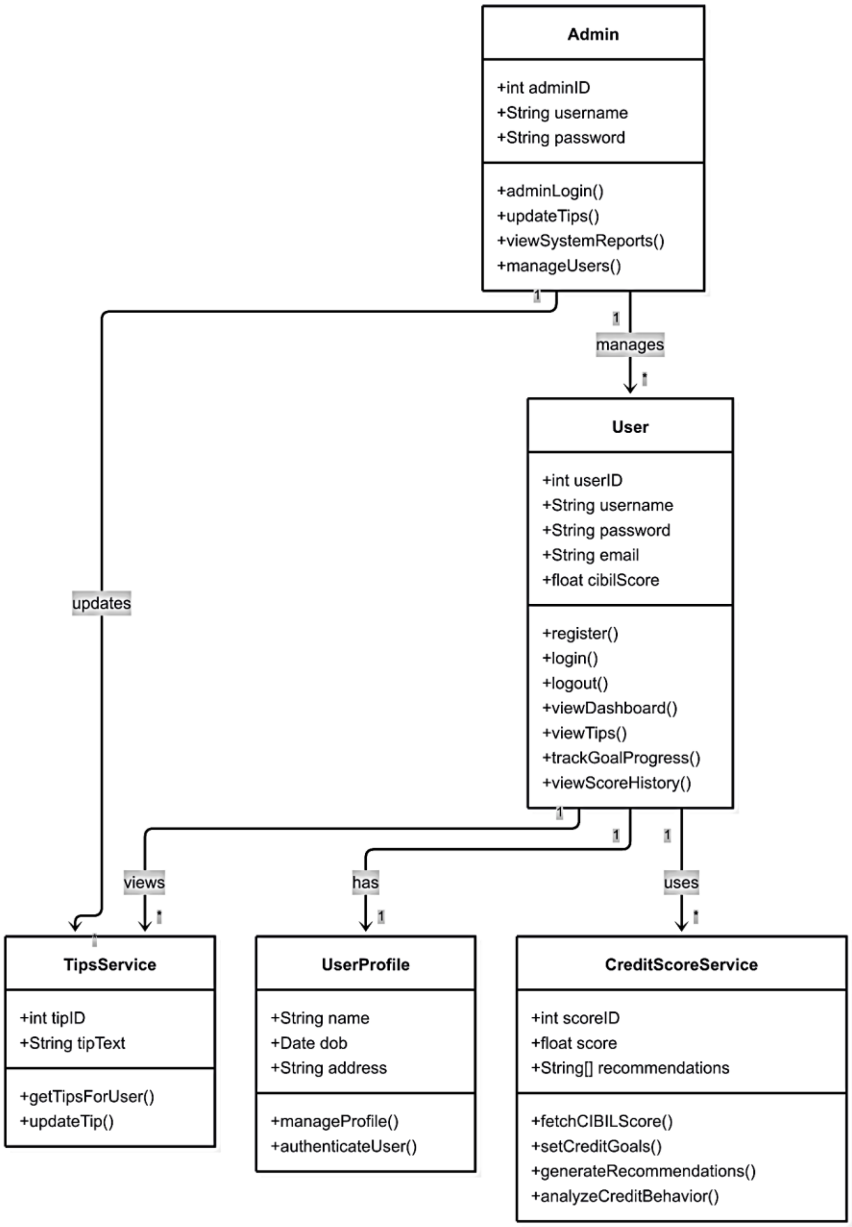
UC 15: View System Reports
Primary Actor: Admin
Precondition: Reports are generated
Trigger: Admin opens reports section

Main Success Scenario:
1. System loads usage stats, user behavior
2. Admin views summary or exports report

UC 16: Manage Users
Primary Actor: Admin
Precondition: Admin is authenticated
Trigger: Admin selects “Manage Users”

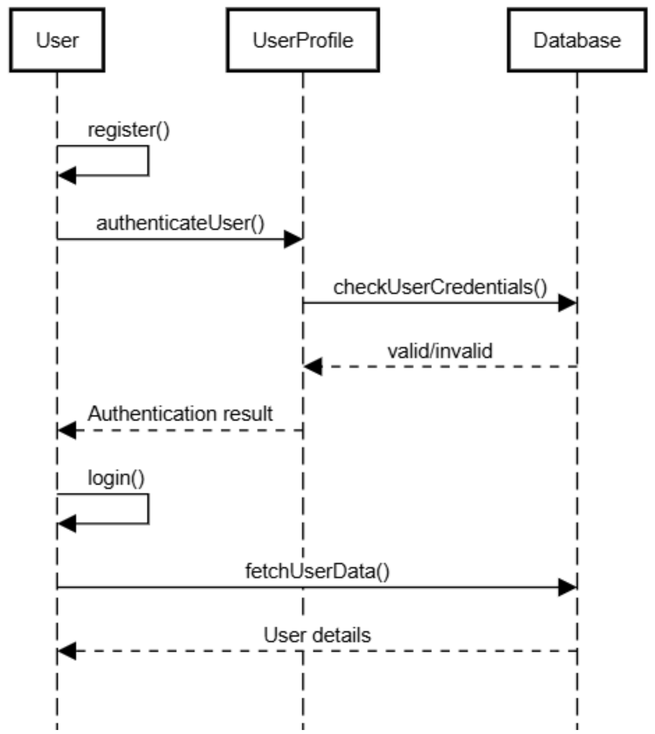
Main Success Scenario:
1. Admin views user list
2. Can activate, deactivate, or delete users

Class Diagram

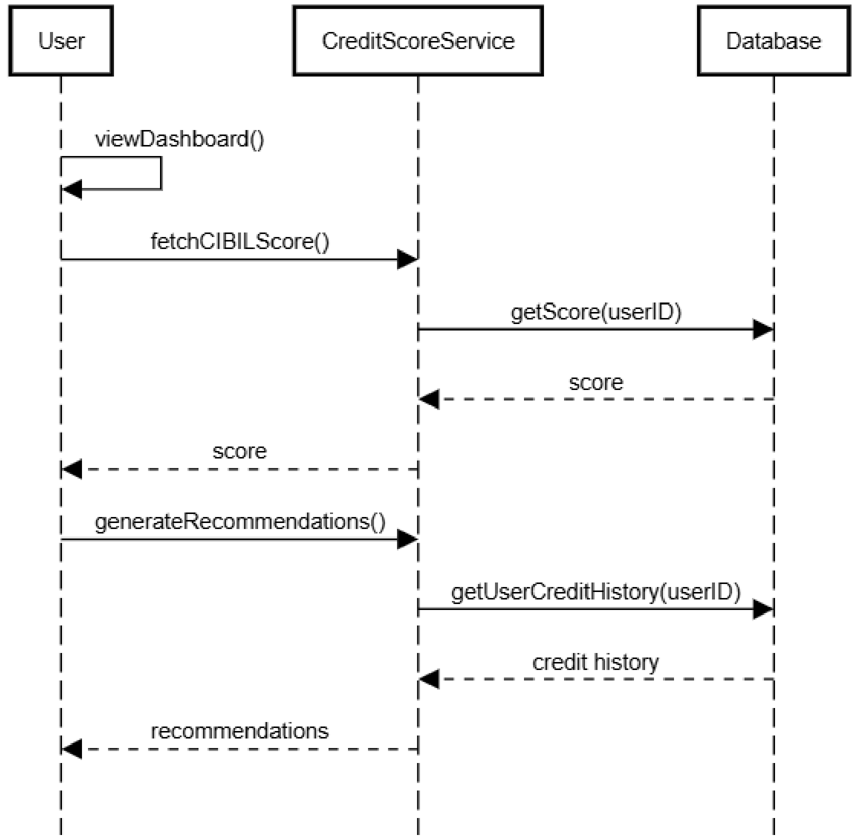


Sequence Diagram

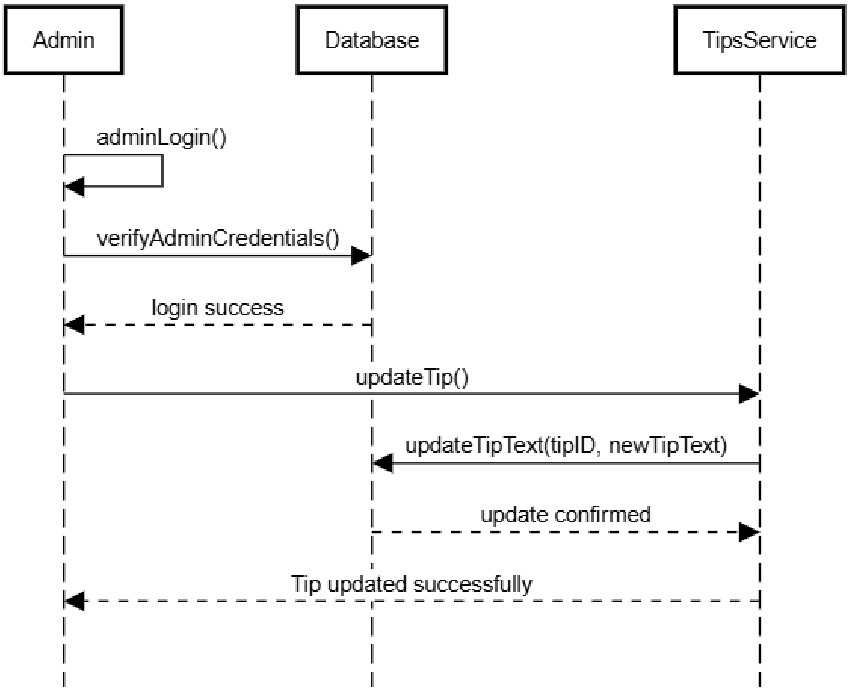
User Registration and Login



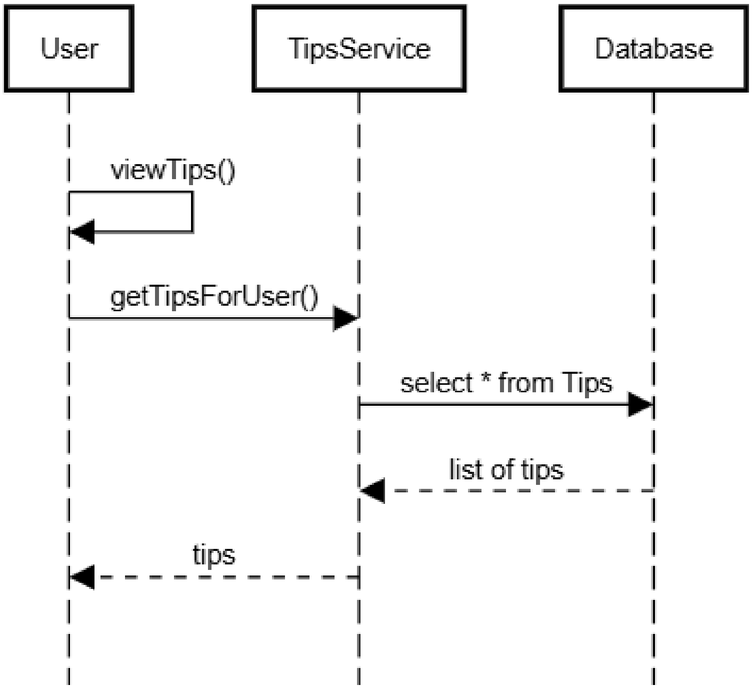
User Views Credit Score and Recommendations

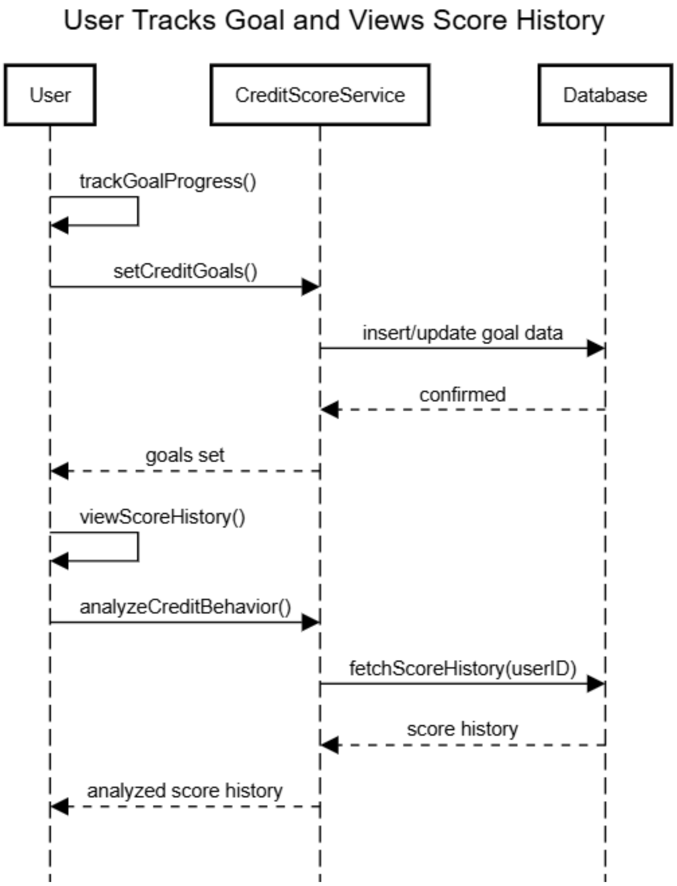


Admin Updates Tips

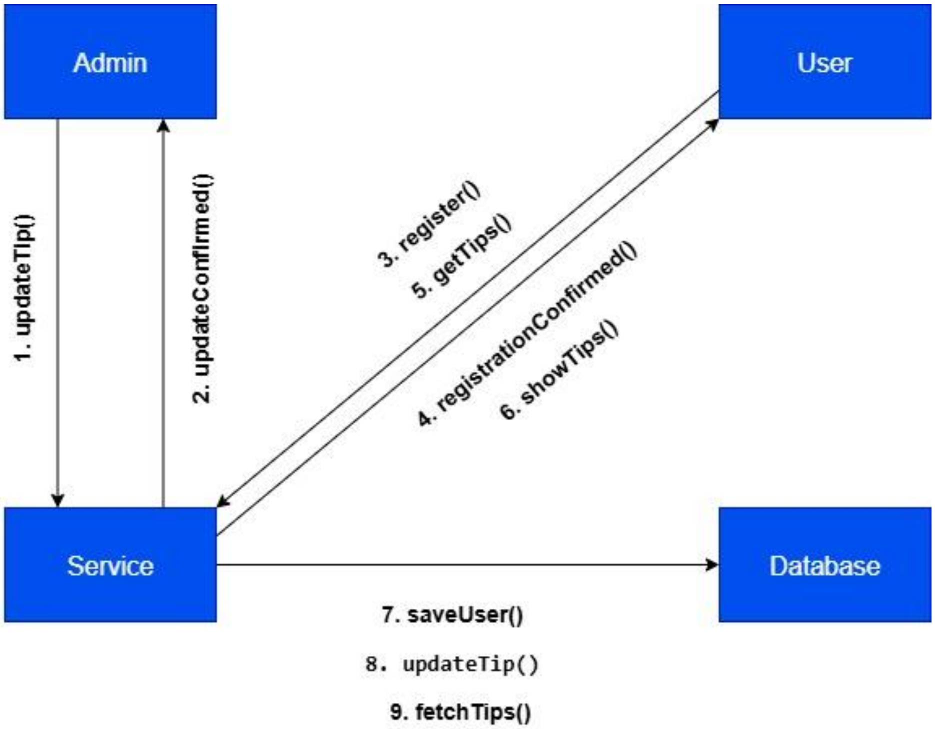


User Views Tips

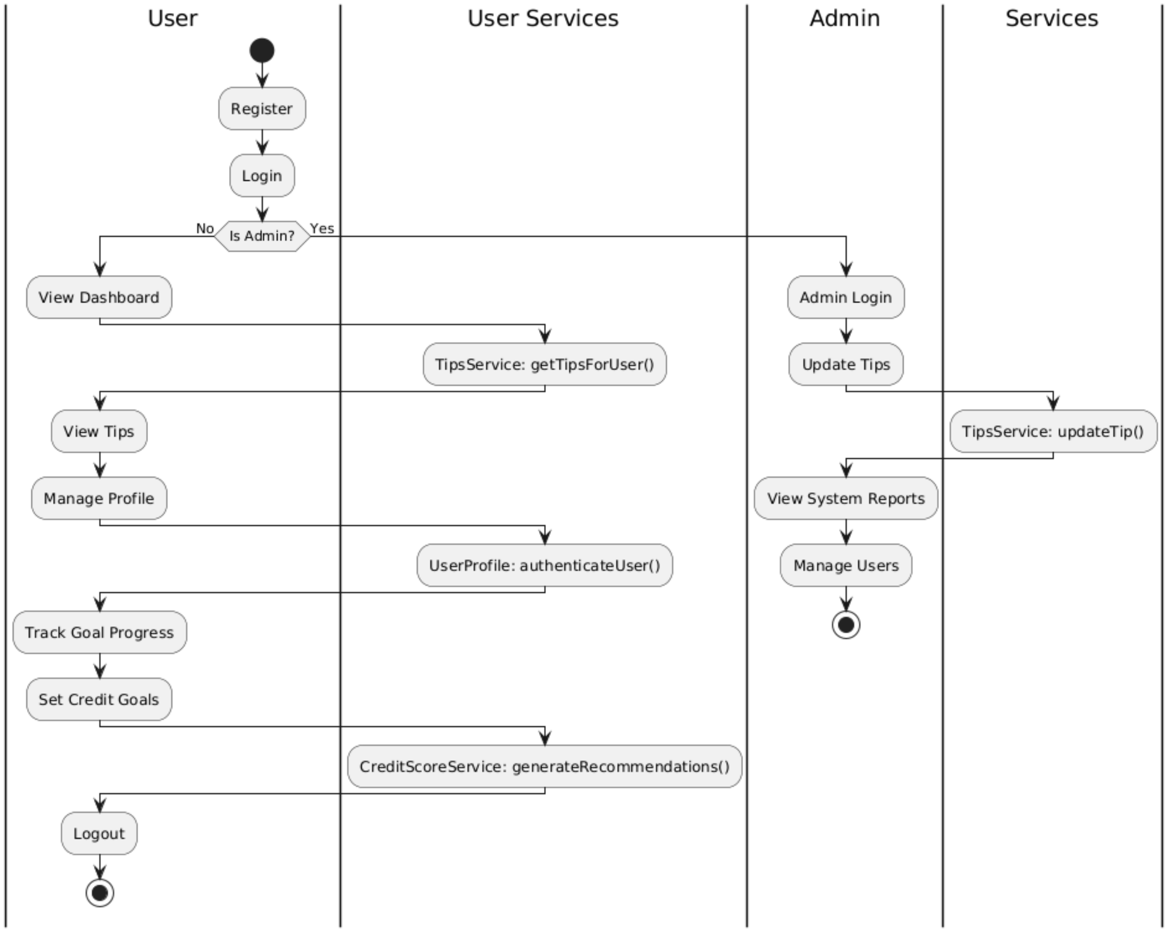




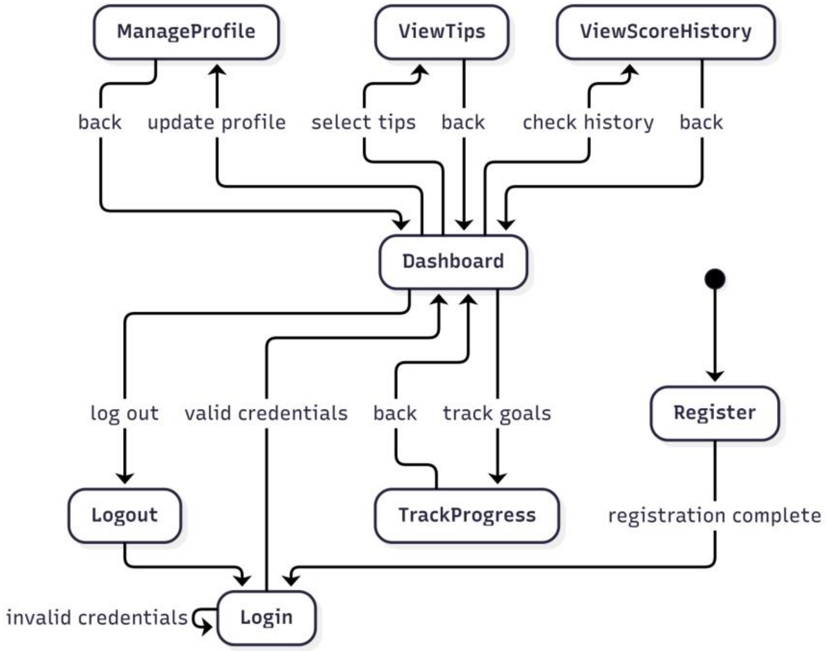
Collaboration Diagram

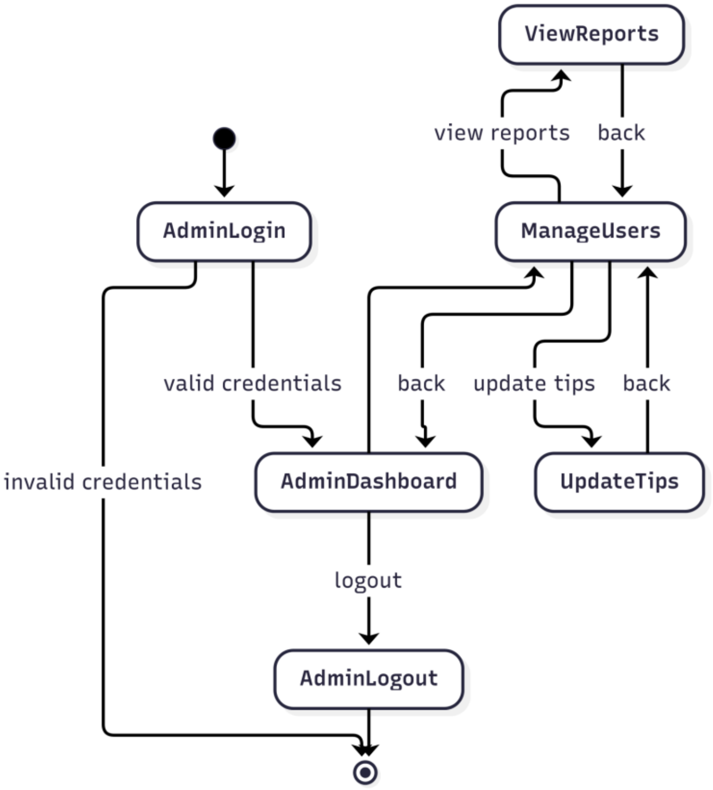


Activity Diagram

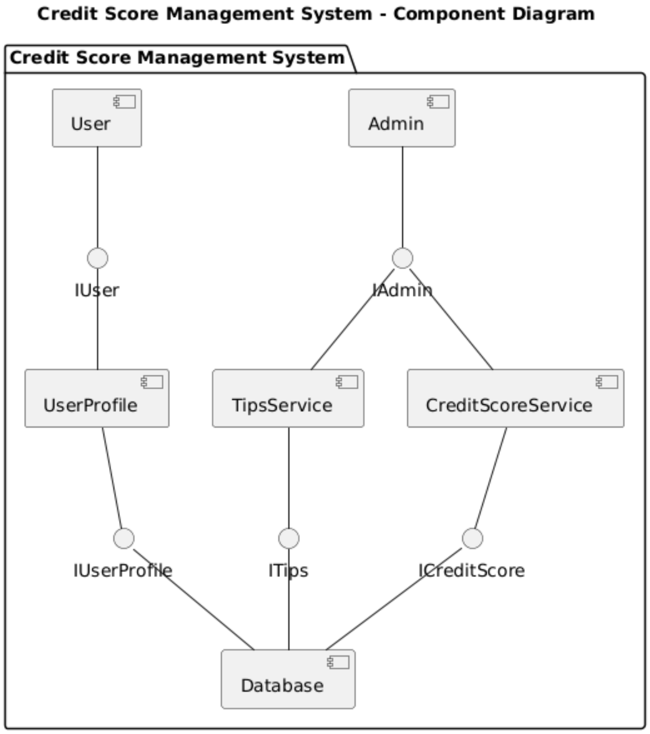


State Diagram

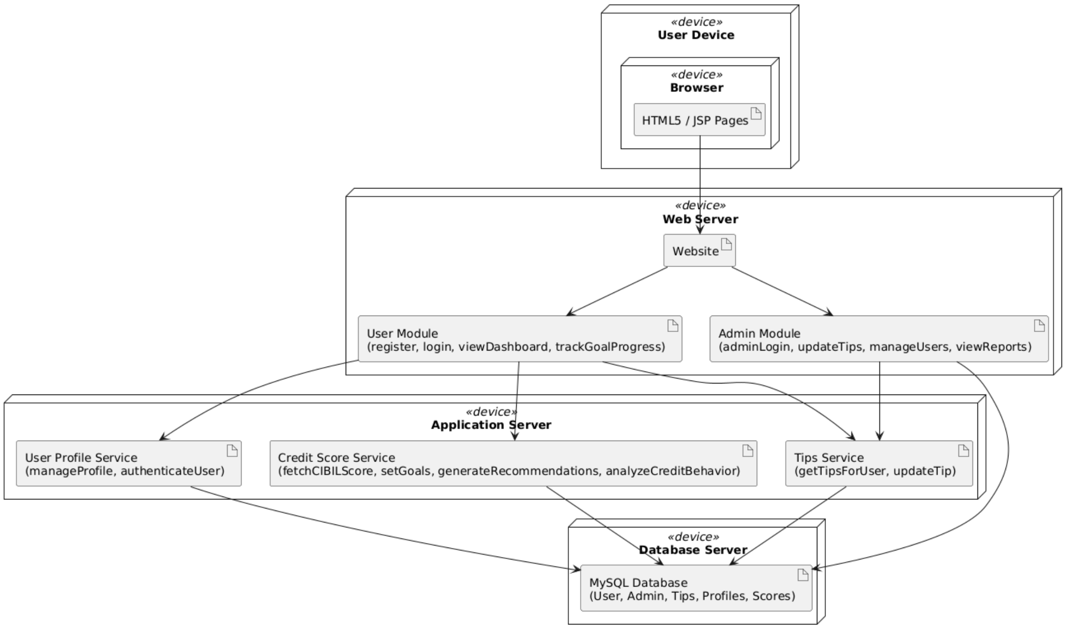




Component Diagram



7. Deployement Diagram


«device»
User Device
«device»
Browser
HTML5 / JSP Pages
«device»
Web Server
Website
User Module
(register, login, viewDashboard, trackGoalProgress)
Admin Module
(adminLogin, updateTips, manageUsers, viewReports)
«device»
Application Server
User Profile Service
(manageProfile, authenticateUser)
Credit Score Service
(fetchCIBILScore, setGoals, generateRecommendations, analyzeCreditBehavior)
Tips Service
(getTipsForUser, updateTip)
«device»
Database Server
MySQL Database
(User, Admin, Tips, Profiles, Scores)

Appendix C: Issues List

The following issues remain open and will be addressed in future revisions:

- **TBD-1:** Integration with real CIBIL API is pending. Currently, the project uses simulated score data.
- **TBD-2:** Decision on internationalization support (multi-currency and multi-language) is pending.
- **TBD-3:** Backup frequency and long-term archival policy need final confirmation.
- **TBD-4:** Requirement for email notifications (alerts/reminders) is under consideration.
- **TBD-5:** Mobile app integration is a possible future enhancement but not part of the current scope.
- **TBD-6:** Security certification requirements (e.g., ISO 27001 compliance) are not yet defined.

SOURCE CODE

Index.html

```
<?php
// No PHP needed here, but renamed to .php for consistency
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CIBIL Score Tracker</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <link rel="stylesheet" href="assets/css/styles.css">
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap"
rel="stylesheet">
</head>
<body class="bg-gradient-to-br from-indigo-50 to-purple-100 min-h-screen font-['Inter']">
    <header class="bg-gradient-to-r from-indigo-600 to-purple-600 text-white py-4 shadow-lg sticky
top-0 z-10 backdrop-blur-md bg-opacity-80">
        <div class="container mx-auto px-4 flex justify-between items-center">
            <div class="flex items-center">
                <img src="./assets/img/1.png" alt="Logo" class="h-10 mr-3">
                <h1 class="text-3xl font-bold tracking-tight">CIBIL Score Tracker</h1>
            </div>
            <nav>
                <ul class="flex space-x-8">
                    <li><a href="index.php" class="hover:text-indigo-200 transition">Home</a></li>
                    <li><a href="pages/user/login.php" class="hover:text-indigo-200
transition">Login</a></li>
                    <li><a href="pages/user/dashboard.php" class="hover:text-indigo-200
transition">Dashboard</a></li>
                    <li><a href="pages/admin/admin-login.php" class="hover:text-indigo-200
transition">Admin</a></li>
                </ul>
            </nav>
        </div>
    </header>

    <section class="hero bg-gradient-to-br from-indigo-700 to-purple-700 text-white min-h-[600px]
flex items-center justify-center">
        <div class="container mx-auto px-4 text-center">
            <h2 class="text-5xl font-bold mb-6 tracking-tight">Master Your Credit Journey</h2>
            <p class="text-xl mb-8 max-w-2xl mx-auto">Track, improve, and manage your CIBIL score with our cutting-edge tools designed for financial empowerment.</p>
            <div class="space-x-4">
                <a href="pages/user/login.php" class="bg-white text-indigo-700 py-3 px-8 rounded-full shadow-lg hover:bg-indigo-100 transition-transform transform hover:scale-105">Get Started</a>
                <a href="pages/user/register.php" class="bg-transparent border-2 border-white py-3 px-8 rounded-full hover:bg-white hover:text-indigo-700 transition-transform transform hover:scale-105">Sign Up</a>
            </div>
        </div>
    </section>

    <section class="features py-16 bg-white">
        <div class="container mx-auto px-4">
            <h2 class="text-4xl font-bold text-center text-gray-800 mb-12">Why Choose CIBIL Score Tracker?</h2>
            <div class="grid grid-cols-1 md:grid-cols-3 gap-8">
                <div class="bg-white bg-opacity-80 backdrop-blur-md p-6 rounded-2xl shadow-xl hover:shadow-2xl transition-shadow">
                    <h3 class="text-xl font-semibold mb-4 text-indigo-600">Real-Time Score Tracking</h3>
                    <p class="text-gray-600">Monitor your CIBIL score with instant updates and detailed history.</p>
                </div>
                <div class="bg-white bg-opacity-80 backdrop-blur-md p-6 rounded-2xl shadow-xl hover:shadow-2xl transition-shadow">
                    <h3 class="text-xl font-semibold mb-4 text-indigo-600">Personalized Goals</h3>
                    <p class="text-gray-600">Set and track credit goals to achieve your financial dreams.</p>
                </div>
                <div class="bg-white bg-opacity-80 backdrop-blur-md p-6 rounded-2xl shadow-xl hover:shadow-2xl transition-shadow">
                    <h3 class="text-xl font-semibold mb-4 text-indigo-600">Expert Recommendations</h3>
                    <p class="text-gray-600">Get tailored tips to boost your credit score.</p>
                </div>
            </div>
        </div>
    </section>

    <section class="testimonials py-16 bg-gradient-to-br from-indigo-50 to-purple-50">
        <div class="container mx-auto px-4">
            <h2 class="text-4xl font-bold text-center text-gray-800 mb-12">What Our Users Say</h2>
            <div class="grid grid-cols-1 md:grid-cols-2 gap-8">
                <div class="bg-white bg-opacity-80 backdrop-blur-md p-6 rounded-2xl shadow-xl">
```

```
                    <p class="text-gray-600 italic">"This app made tracking my credit score so easy! The tips helped me improve my score by 50 points!"</p>
                    <p class="mt-4 font-semibold text-indigo-600">- Priya S.</p>
                </div>
                <div class="bg-white bg-opacity-80 backdrop-blur-md p-6 rounded-2xl shadow-xl">
                    <p class="text-gray-600 italic">"The dashboard is intuitive, and the goal-setting feature keeps me motivated."</p>
                    <p class="mt-4 font-semibold text-indigo-600">- Rohan K.</p>
                </div>
            </div>
        </div>
    </section>

    <footer class="bg-gray-900 text-white py-8">
        <div class="container mx-auto px-4">
            <div class="grid grid-cols-1 md:grid-cols-3 gap-8">
                <div>
                    <h3 class="text-xl font-semibold mb-4">CIBIL Score Tracker</h3>
                    <p class="text-gray-400">Empowering you to take control of your financial future.</p>
                </div>
                <div>
                    <h3 class="text-xl font-semibold mb-4">Links</h3>
                    <ul class="space-y-2">
                        <li><a href="index.php" class="text-gray-400 hover:text-indigo-300 transition">Home</a></li>
                        <li><a href="pages/user/login.php" class="text-gray-400 hover:text-indigo-300 transition">Login</a></li>
                        <li><a href="pages/user/register.php" class="text-gray-400 hover:text-indigo-300 transition">Register</a></li>
                    </ul>
                </div>
                <div>
                    <h3 class="text-xl font-semibold mb-4">Contact</h3>
                    <p class="text-gray-400">Email: support@cibiltracker.com</p>
                    <p class="text-gray-400">Phone: +91 123-456-7890</p>
                </div>
            </div>
            <p class="text-center text-gray-400 mt-8">&copy; 2025 CIBIL Score Tracker. All rights reserved.</p>
        </div>
    </footer>

    <script src="assets/js/script.js"></script>
</body>
</html>
```

Dashboard.php

```
<?php
require_once '../../config.php';

session_start();

// Check if the user is logged in, if not then redirect him to login page
if (!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true || $_SESSION["role"] !==
'user') {
    header("location: login.php");
    exit;
}

// Fetch tips from DB
$tips_stmt = $pdo->query("SELECT tip_text FROM tips");
$tips = $tips_stmt->fetchAll(PDO::FETCH_COLUMN);

// Fetch score history from DB
$history_stmt = $pdo->prepare("SELECT score, fetched_at AS date FROM score_history WHERE
user_id = :user_id ORDER BY fetched_at DESC");
$history_stmt->execute(['user_id' => $_SESSION["id"]]);
$history = $history_stmt->fetchAll(PDO::FETCH_ASSOC);

// Process goal form (simple, store in session for demo; in real, add goals table)
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['goal'])) {
    $_SESSION['goal'] = $_POST['goal']; // Simulated
    header("location: dashboard.php");
    exit;
}

// Process profile update (update DB)
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['name']) &&
isset($_POST['email'])) {
    $sql = "UPDATE users SET username = :username, email = :email WHERE id = :id";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(['username' => $_POST['name'], 'email' => $_POST['email'], 'id' =>
$_SESSION["id"]]);
    $_SESSION["username"] = $_POST['name'];
    header("location: dashboard.php");
    exit;
}

// Simulated progress (hardcoded 70%)
```

```
$progress = 70;
$goal = isset($_SESSION['goal']) ? $_SESSION['goal'] : '';
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard - CIBIL Score Tracker</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <link rel="stylesheet" href="../../assets/css/styles.css">
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap"
rel="stylesheet">
</head>
<body class="bg-gradient-to-br from-indigo-50 to-purple-100 min-h-screen font-['Inter']">
    <header class="bg-gradient-to-r from-indigo-600 to-purple-600 text-white py-4 shadow-lg sticky
top-0 z-10 backdrop-blur-md bg-opacity-80">
        <div class="container mx-auto px-4 flex justify-between items-center">
            <div class="flex items-center">
                <img src="../../assets/img/1.png" alt="Logo" class="h-10 mr-3">
                <h1 class="text-3xl font-bold tracking-tight">CIBIL Score Tracker</h1>
            </div>
            <nav>
                <ul class="flex space-x-8">
                    <li><a href="../../index.php" class="hover:text-indigo-200 transition">Home</a></li>
                    <li><a href="dashboard.php" class="hover:text-indigo-200
transition">Dashboard</a></li>
                    <li><a href="../../logout.php" class="hover:text-indigo-200
transition">Logout</a></li>
                </ul>
            </nav>
        </div>
    </header>

    <main class="container mx-auto px-4 py-12">
        <h2 class="text-4xl font-bold mb-8 text-gray-800">Welcome, <?php echo
htmlspecialchars($_SESSION["username"]); ?></h2>
        <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
            <!-- Fetch CIBIL Score -->
            <div class="bg-white bg-opacity-80 backdrop-blur-md p-6 rounded-2xl shadow-xl
hover:shadow-2xl transition-shadow transform hover:scale-105">
                <h3 class="text-xl font-semibold mb-4 text-indigo-600">Fetch CIBIL Score</h3>
                <button id="fetchScoreBtn" class="bg-indigo-600 text-white py-2 px-6 rounded-lg
hover:bg-indigo-700 transition-transform transform hover:scale-105 mb-4">Fetch Score</button>
                <p class="text-5xl font-bold text-gray-800" id="score">--</p>
            </div>

            <!-- Manage Profile & Settings -->
            <div class="bg-white bg-opacity-80 backdrop-blur-md p-6 rounded-2xl shadow-xl hover:shadow-2xl transition-shadow transform hover:scale-105">
                <h3 class="text-xl font-semibold mb-4 text-indigo-600">Manage Profile & Settings</h3>
                <form method="post" class="space-y-4">
                    <input type="text" name="name" placeholder="Name" value="<?php echo htmlspecialchars($_SESSION["username"]); ?>" class="w-full p-3 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-indigo-500 bg-white bg-opacity-50">
                    <!-- <input type="email" name="email" placeholder="Email" value="<?php echo htmlspecialchars($_SESSION["email"]); ?>" class="w-full p-3 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-indigo-500 bg-white bg-opacity-50"> -->
                    <button type="submit" class="bg-indigo-600 text-white py-2 px-6 rounded-lg hover:bg-indigo-700 transition-transform transform hover:scale-105">Save</button>
                </form>
            </div>

            <!-- View Tips for Improvement -->
            <div class="bg-white bg-opacity-80 backdrop-blur-md p-6 rounded-2xl shadow-xl hover:shadow-2xl transition-shadow transform hover:scale-105">
                <h3 class="text-xl font-semibold mb-4 text-indigo-600">Tips for Improvement</h3>
                <ul class="list-disc pl-5 space-y-2 text-gray-600">
                    <?php foreach ($tips as $tip): ?>
                        <li><?php echo htmlspecialchars($tip); ?></li>
                    <?php endforeach; ?>
                </ul>
            </div>

            <!-- Track Goal Progress -->
            <div class="bg-white bg-opacity-80 backdrop-blur-md p-6 rounded-2xl shadow-xl hover:shadow-2xl transition-shadow transform hover:scale-105">
                <h3 class="text-xl font-semibold mb-4 text-indigo-600">Track Goal Progress</h3>
                <progress value="<?php echo $progress; ?>" max="100" class="w-full h-4 mb-2 rounded bg-indigo-200"></progress>
                <p class="text-gray-600"><?php echo $progress; ?>% towards goal</p>
            </div>

            <!-- Set Credit Goals -->
            <div class="bg-white bg-opacity-80 backdrop-blur-md p-6 rounded-2xl shadow-xl hover:shadow-2xl transition-shadow transform hover:scale-105">
                <h3 class="text-xl font-semibold mb-4 text-indigo-600">Set Credit Goals</h3>
                <form method="post" class="space-y-4">
```

```
                <input type="text" name="goal" placeholder="Goal (e.g., Reach 750)" value="<?php echo htmlspecialchars($goal); ?>" class="w-full p-3 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-indigo-500 bg-white bg-opacity-50">
                <button type="submit" class="bg-indigo-600 text-white py-2 px-6 rounded-lg hover:bg-indigo-700 transition-transform transform hover:scale-105">Save</button>
            </form>
        </div>

        <!-- Generate Recommendations -->
        <div class="bg-white bg-opacity-80 backdrop-blur-md p-6 rounded-2xl shadow-xl hover:shadow-2xl transition-shadow transform hover:scale-105">
            <h3 class="text-xl font-semibold mb-4 text-indigo-600">Generate Recommendations</h3>
            <button id="recBtn" class="bg-indigo-600 text-white py-2 px-6 rounded-lg hover:bg-indigo-700 transition-transform transform hover:scale-105 mb-4">Generate</button>
            <p id="recs" class="italic text-gray-600">--</p>
        </div>

        <!-- Analyze Credit Behavior -->
        <div class="bg-white bg-opacity-80 backdrop-blur-md p-6 rounded-2xl shadow-xl hover:shadow-2xl transition-shadow transform hover:scale-105">
            <h3 class="text-xl font-semibold mb-4 text-indigo-600">Analyze Credit Behavior</h3>
            <div class="h-40 bg-gray-100 rounded flex items-center justify-center text-gray-500">Chart Placeholder</div>
        </div>

        <!-- View Score History -->
        <div class="bg-white bg-opacity-80 backdrop-blur-md p-6 rounded-2xl shadow-xl hover:shadow-2xl transition-shadow col-span-1 md:col-span-2 lg:col-span-3">
            <h3 class="text-xl font-semibold mb-4 text-indigo-600">Score History</h3>
            <button id="historyBtn" class="bg-indigo-600 text-white py-2 px-6 rounded-lg hover:bg-indigo-700 transition-transform transform hover:scale-105 mb-4">Add Sample Score</button>
            <table id="scoreHistory" class="w-full border-collapse">
                <thead>
                    <tr class="bg-indigo-100">
                        <th class="p-3 text-left text-indigo-600">Date</th>
                        <th class="p-3 text-left text-indigo-600">Score</th>
                    </tr>
                </thead>
                <tbody>
                    <?php foreach ($history as $row): ?>
                        <tr>
                            <td class="p-3 border border-indigo-100"><?php echo htmlspecialchars($row['date']); ?></td>
```

```
                                <td class="p-3 border border-indigo-100"><?php echo
htmlspecialchars($row['score']); ?></td>
                            </tr>
                        <?php endforeach; ?>
                    </tbody>
                </table>
            </div>
        </div>
    </main>
    <script src="../../assets/js/script.js"></script>
</body>
</html>
```

Login.php

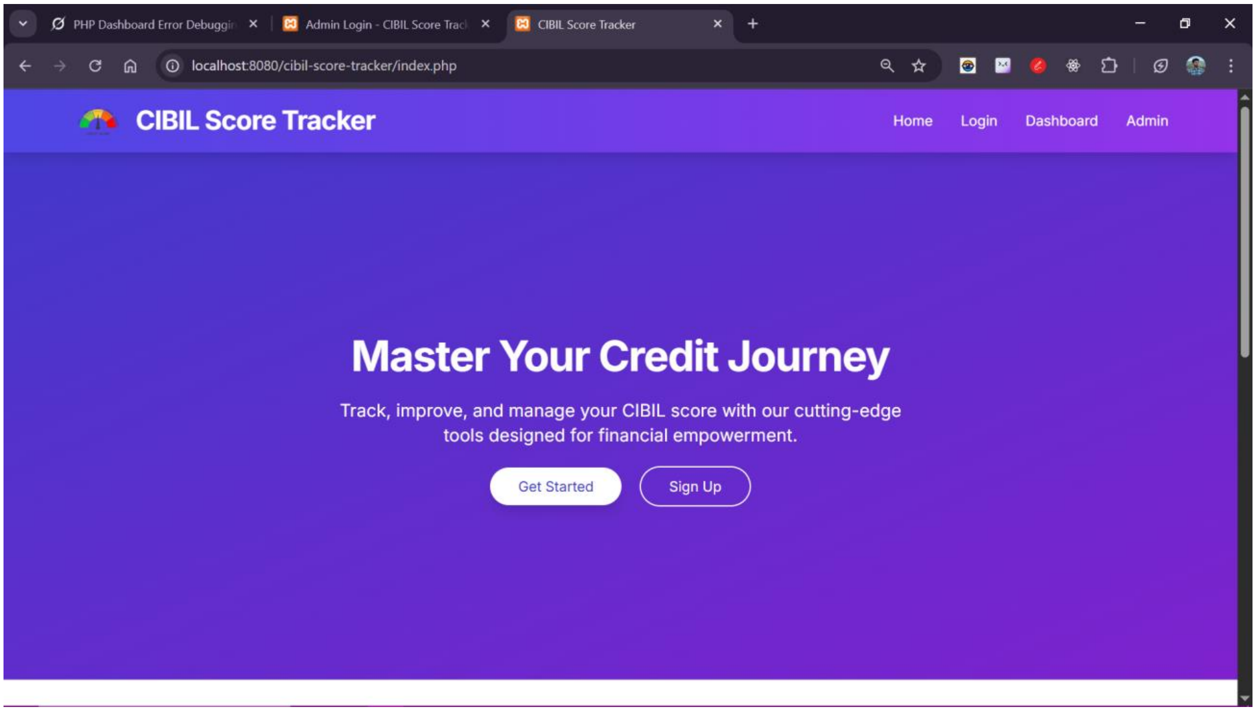
```
<?php
require_once '../../config.php';

session_start();

// Check if the user is already logged in, if yes then redirect him to dashboard page
if (isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true) {
    if ($_SESSION["role"] === 'admin') {
        header("location: ../../pages/admin/admin-dashboard.php");
    } else {
        header("location: dashboard.php");
    }
    exit;
}

// Define variables and initialize with empty values
$email = $password = "";
$email_err = $password_err = $login_err = "";

// Processing form data when form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {

    // Check if email is empty
    if (empty(trim($_POST["email"]))) {
        $email_err = "Please enter email.";
    } else {
        $email = trim($_POST["email"]);
    }

    // Check if password is empty
    if (empty(trim($_POST["password"]))) {
        $password_err = "Please enter your password.";
    } else {
        $password = trim($_POST["password"]);
    }

    // Validate credentials
    if (empty($email_err) && empty($password_err)) {
        // Prepare a select statement
        $sql = "SELECT id, username, email, password_hash, role FROM users WHERE email = 
:email";
        $stmt = $pdo->prepare($sql);
        $stmt->execute(['email' => $email]);
        $user = $stmt->fetch();

        if ($user && password_verify($password, $user['password_hash'])) {
            // Password is correct, so start a new session
            session_start();

            // Store data in session variables
            $_SESSION["loggedin"] = true;
            $_SESSION["id"] = $user['id'];
            $_SESSION["username"] = $user['username'];
            $_SESSION["role"] = $user['role'];

            // Redirect user to dashboard page
            if ($_SESSION["role"] === 'admin') {
                header("location: ../../pages/admin/admin-dashboard.php");
            } else {
                header("location: dashboard.php");
            }
            exit;
        } else {
            // Password is not valid, display a generic error message
            $login_err = "Invalid email or password.";
        }
    }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login - CIBIL Score Tracker</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <link rel="stylesheet" href="../../assets/css/styles.css">
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap" rel="stylesheet">
</head>
<body class="bg-gradient-to-br from-indigo-50 to-purple-100 min-h-screen flex items-center justify-center font-['Inter']">
    <div class="max-w-md w-full bg-white bg-opacity-80 backdrop-blur-md p-8 rounded-2xl shadow-xl">
        <h2 class="text-3xl font-bold mb-6 text-center text-gray-800">Sign In</h2>
        <span class="text-red-500 text-sm"><?php echo $login_err; ?></span>
        <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post" class="space-y-6">
            <div class="relative">
                <label class="block text-gray-700 mb-2" for="email">Email</label>
                <input type="email" name="email" id="email" class="w-full p-3 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-indigo-500 bg-white bg-opacity-50 <?php echo (!empty($email_err)) ? 'border-red-500' : "; ?>" value="<?php echo $email; ?>" placeholder="Enter email">
                <span class="text-red-500 text-sm"><?php echo $email_err; ?></span>
            </div>
            <div class="relative">
                <label class="block text-gray-700 mb-2" for="password">Password</label>
                <input type="password" name="password" id="password" class="w-full p-3 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-indigo-500 bg-white bg-opacity-50 <?php echo (!empty($password_err)) ? 'border-red-500' : "; ?>" placeholder="Enter password">
                <span class="text-red-500 text-sm"><?php echo $password_err; ?></span>
            </div>
            <button type="submit" class="w-full bg-indigo-600 text-white py-3 rounded-lg hover:bg-indigo-700 transition-transform transform hover:scale-105">Login</button>
        </form>
        <p class="text-center mt-6 text-gray-600">Don't have an account? <a href="register.php" class="text-indigo-600 hover:underline">Register</a></p>
    </div>
    <script src="../../assets/js/script.js"></script>
</body>
</html>
```

Register.php

```
<?php
require_once '../../config.php';

// Define variables and initialize with empty values
```

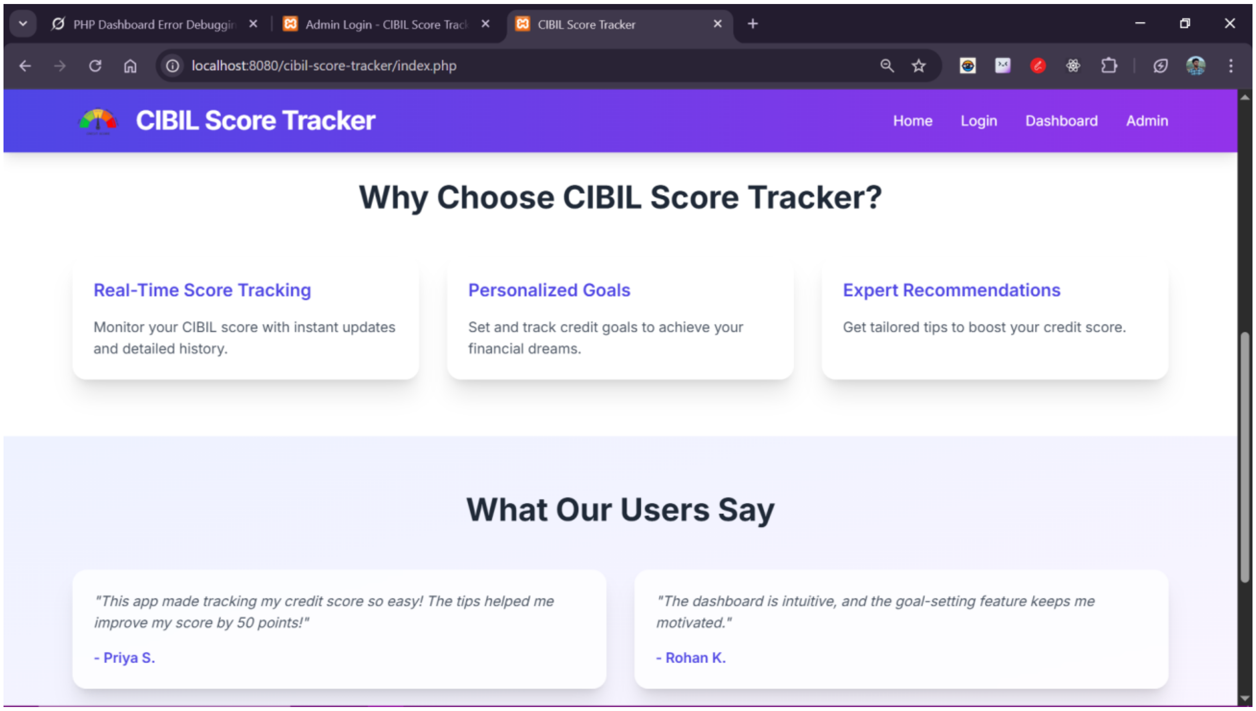
```
$username = $email = $password = $confirm_password = "";
$username_err = $email_err = $password_err = $confirm_password_err = "";

// Processing form data when form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {

    // Validate username
    if (empty(trim($_POST["username"]))) {
        $username_err = "Please enter a username.";
    } elseif (!preg_match('/^[a-zA-Z0-9_]+$/', trim($_POST["username"]))) {
        $username_err = "Username can only contain letters, numbers, and underscores.";
    } else {
        $sql = "SELECT id FROM users WHERE username = :username";
        $stmt = $pdo->prepare($sql);
        $stmt->execute(['username' => trim($_POST["username"])]);
        if ($stmt->rowCount() > 0) {
            $username_err = "This username is already taken.";
        } else {
            $username = trim($_POST["username"]);
        }
    }

    // Validate email
    if (empty(trim($_POST["email"]))) {
        $email_err = "Please enter an email.";
    } elseif (!filter_var(trim($_POST["email"]), FILTER_VALIDATE_EMAIL)) {
        $email_err = "Invalid email format.";
    } else {
        $sql = "SELECT id FROM users WHERE email = :email";
        $stmt = $pdo->prepare($sql);
        $stmt->execute(['email' => trim($_POST["email"])]);
        if ($stmt->rowCount() > 0) {
            $email_err = "This email is already taken.";
        } else {
            $email = trim($_POST["email"]);
        }
    }

    // Validate password
    if (empty(trim($_POST["password"]))) {
        $password_err = "Please enter a password.";
    } elseif (strlen(trim($_POST["password"])) < 6) {
        $password_err = "Password must have at least 6 characters.";
    } else {
        $password = trim($_POST["password"]);
    }

    // Validate confirm password
    if (empty(trim($_POST["confirm_password"]))) {
        $confirm_password_err = "Please confirm password.";
    } else {
        $confirm_password = trim($_POST["confirm_password"]);
        if (empty($password_err) && ($password != $confirm_password)) {
            $confirm_password_err = "Password did not match.";
        }
    }

    // Check input errors before inserting in database
    if (empty($username_err) && empty($email_err) && empty($password_err) &&
empty($confirm_password_err)) {
        $sql = "INSERT INTO users (username, email, password_hash, role) VALUES (:username,
:email, :password_hash, 'user')";
        $stmt = $pdo->prepare($sql);
        $password_hash = password_hash($password, PASSWORD_DEFAULT);
        if ($stmt->execute(['username' => $username, 'email' => $email, 'password_hash' =>
$password_hash])) {
            header("location: login.php");
            exit;
        } else {
            echo "Oops! Something went wrong. Please try again later.";
        }
    }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Register - CIBIL Score Tracker</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <link rel="stylesheet" href="../../assets/css/styles.css">
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap"
rel="stylesheet">
</head>
<body class="bg-gradient-to-br from-indigo-50 to-purple-100 min-h-screen flex items-center
justify-center font-['Inter']">
    <div class="max-w-md w-full bg-white bg-opacity-80 backdrop-blur-md p-8 rounded-2xl
shadow-xl">
        <h2 class="text-3xl font-bold mb-6 text-center text-gray-800">Create Your Account</h2>
        <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post"
class="space-y-6">
            <div class="relative">
                <label class="block text-gray-700 mb-2" for="username">Username</label>
                <input type="text" name="username" id="username" class="w-full p-3 border border-
gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-indigo-500 bg-white bg-opacity-50
<?php echo (!empty($username_err)) ? 'border-red-500' : ''; ?>" value="<?php echo $username; ?>"
placeholder="Enter username">
                <span class="text-red-500 text-sm"><?php echo $username_err; ?></span>
            </div>
            <div class="relative">
                <label class="block text-gray-700 mb-2" for="email">Email</label>
                <input type="email" name="email" id="email" class="w-full p-3 border border-gray-300
rounded-lg focus:outline-none focus:ring-2 focus:ring-indigo-500 bg-white bg-opacity-50 <?php
echo (!empty($email_err)) ? 'border-red-500' : ''; ?>" value="<?php echo $email; ?>"
placeholder="Enter email">
                <span class="text-red-500 text-sm"><?php echo $email_err; ?></span>
            </div>
            <div class="relative">
                <label class="block text-gray-700 mb-2" for="password">Password</label>
                <input type="password" name="password" id="password" class="w-full p-3 border
border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-indigo-500 bg-white bg-
opacity-50 <?php echo (!empty($password_err)) ? 'border-red-500' : ''; ?>" placeholder="Enter
password">
                <span class="text-red-500 text-sm"><?php echo $password_err; ?></span>
            </div>
            <div class="relative">
                <label class="block text-gray-700 mb-2" for="confirm_password">Confirm
Password</label>
                <input type="password" name="confirm_password" id="confirm_password" class="w-
full p-3 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-indigo-500
bg-white bg-opacity-50 <?php echo (!empty($confirm_password_err)) ? 'border-red-500' : ''; ?>"
placeholder="Confirm password">
                <span class="text-red-500 text-sm"><?php echo $confirm_password_err; ?></span>
            </div>
            <button type="submit" class="w-full bg-indigo-600 text-white py-3 rounded-lg hover:bg-
indigo-700 transition-transform transform hover:scale-105">Register</button>
        </form>
        <p class="text-center mt-6 text-gray-600">Already have an account? <a href="login.php"
class="text-indigo-600 hover:underline">Login</a></p>
    </div>
    <script src="../../assets/js/script.js"></script>
</body>
</html>
```

DEMONSTRATION


PHP Dashboard Error Debuggi
Admin Login - CIBIL Score Trac
CIBIL Score Tracker
localhost:8080/cibil-score-tracker/index.php
CIBIL Score Tracker
Home
Login
Dashboard
Admin
Get Started
Sign Up

Master Your Credit Journey

Track, improve, and manage your CIBIL score with our cutting-edge tools designed for financial empowerment.


PHP Dashboard Error Debuggi
Admin Login - CIBIL Score Trac
CIBIL Score Tracker
localhost:8080/cibil-score-tracker/index.php
CIBIL Score Tracker
Home
Login
Dashboard
Admin

Why Choose CIBIL Score Tracker?

Real-Time Score Tracking

Monitor your CIBIL score with instant updates and detailed history.

Personalized Goals

Set and track credit goals to achieve your financial dreams.

Expert Recommendations

Get tailored tips to boost your credit score.

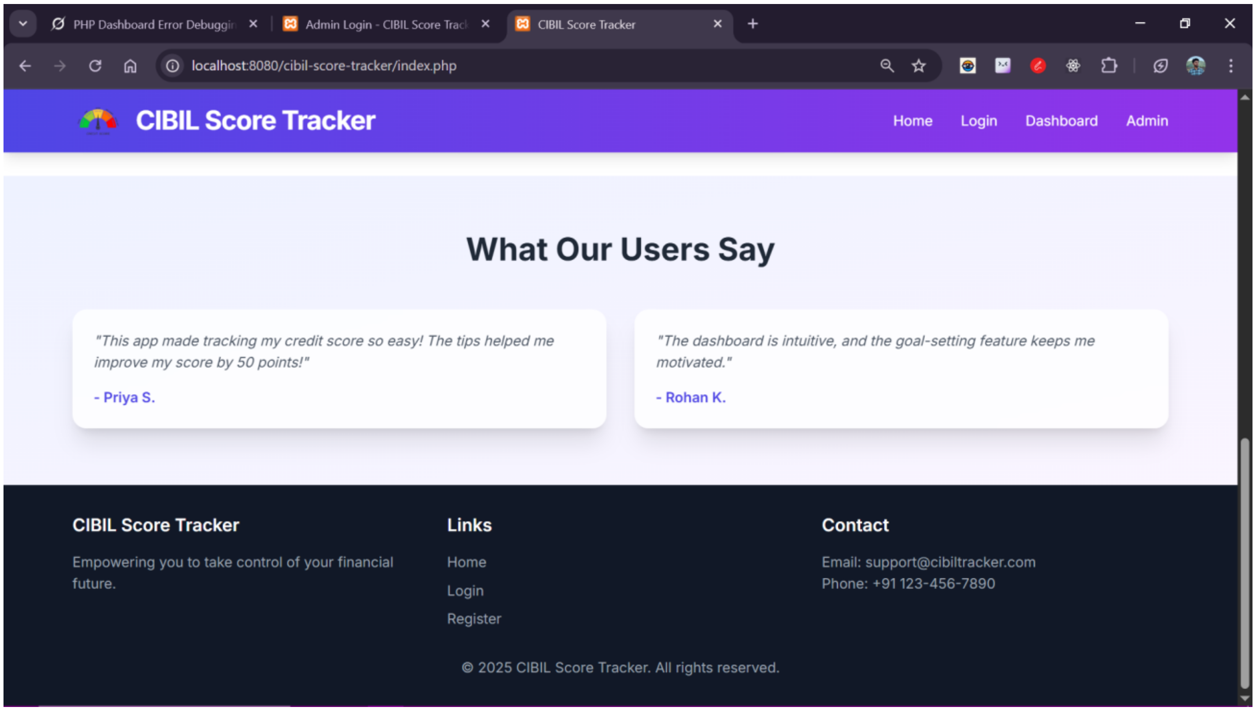
What Our Users Say

"This app made tracking my credit score so easy! The tips helped me improve my score by 50 points!"

- Priya S.

"The dashboard is intuitive, and the goal-setting feature keeps me motivated."

- Rohan K.


PHP Dashboard Error Debuggin
Admin Login - CIBIL Score Track
CIBIL Score Tracker
localhost:8080/cibil-score-tracker/index.php
Home
Login
Dashboard
Admin

CIBIL Score Tracker

What Our Users Say

"This app made tracking my credit score so easy! The tips helped me improve my score by 50 points!"

- Priya S.

"The dashboard is intuitive, and the goal-setting feature keeps me motivated."

- Rohan K.

CIBIL Score Tracker

Empowering you to take control of your financial future.

Links

Home

Login

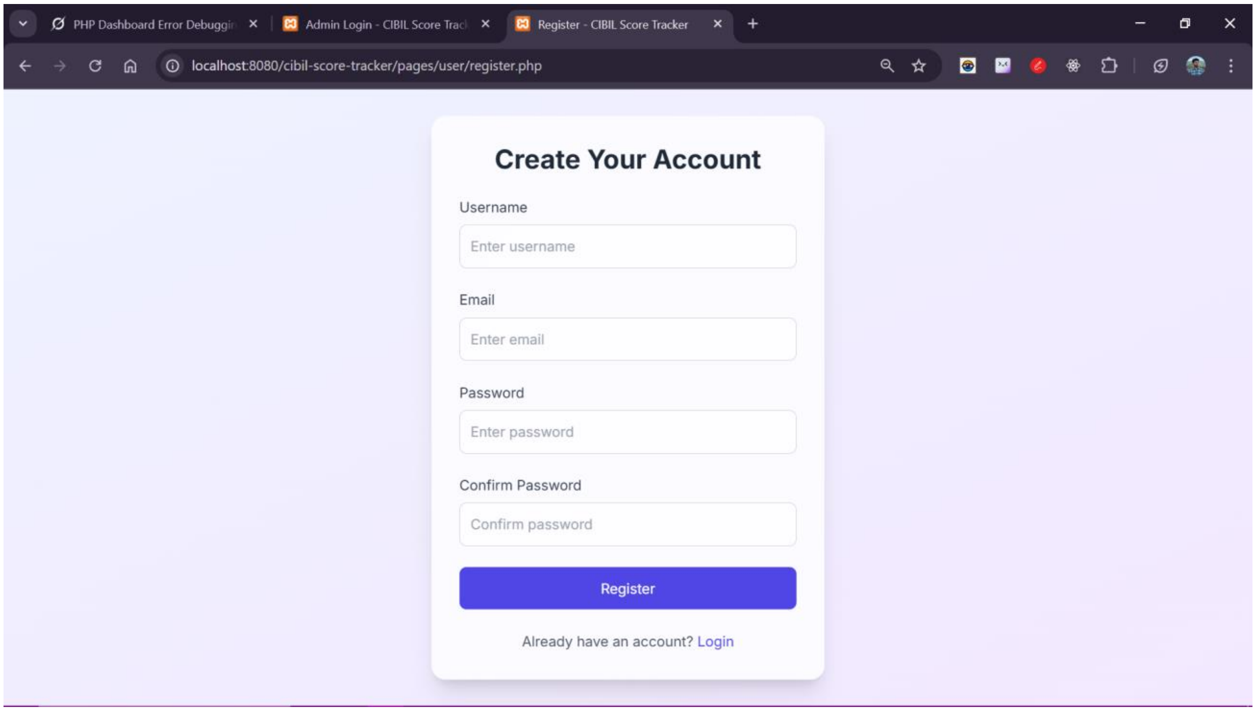
Register

Contact

Email: support@cibiltracker.com

Phone: +91 123-456-7890

© 2025 CIBIL Score Tracker. All rights reserved.


PHP Dashboard Error Debuggin
Admin Login - CIBIL Score Track
Register - CIBIL Score Tracker
localhost:8080/cibil-score-tracker/pages/user/register.php

Create Your Account

Username

Enter username

Email

Enter email

Password

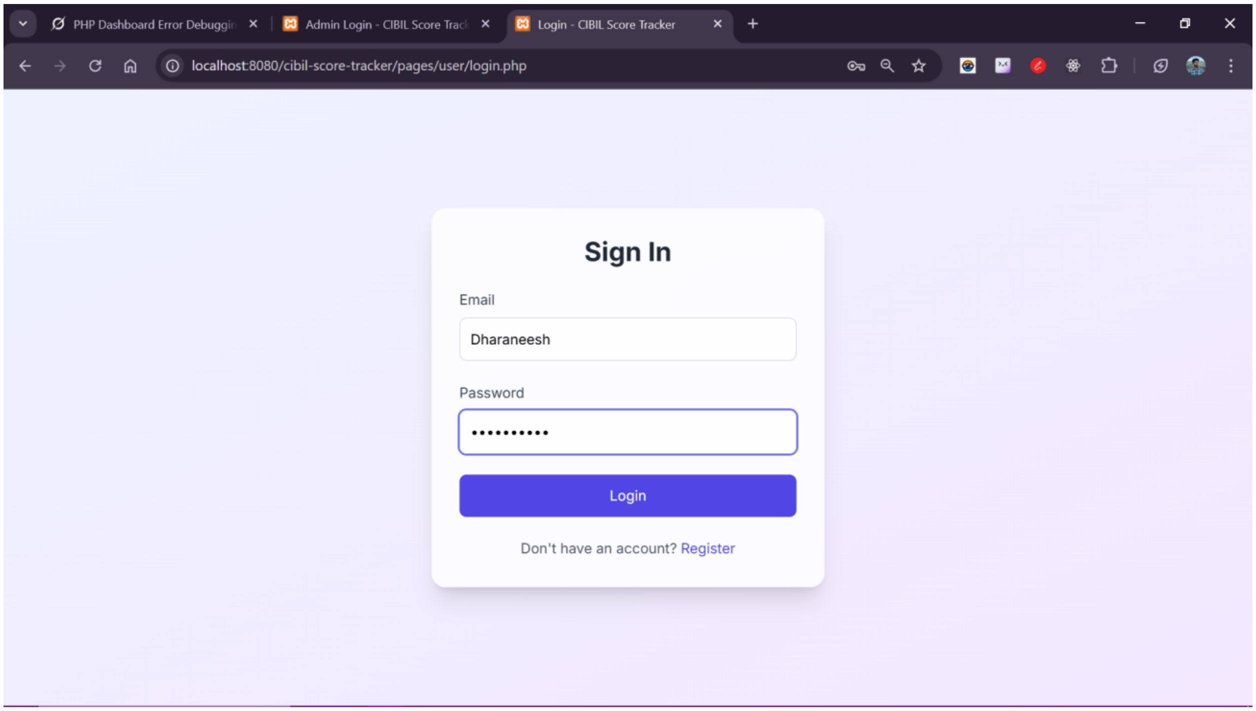
Enter password

Confirm Password

Confirm password

Register

Already have an account? Login

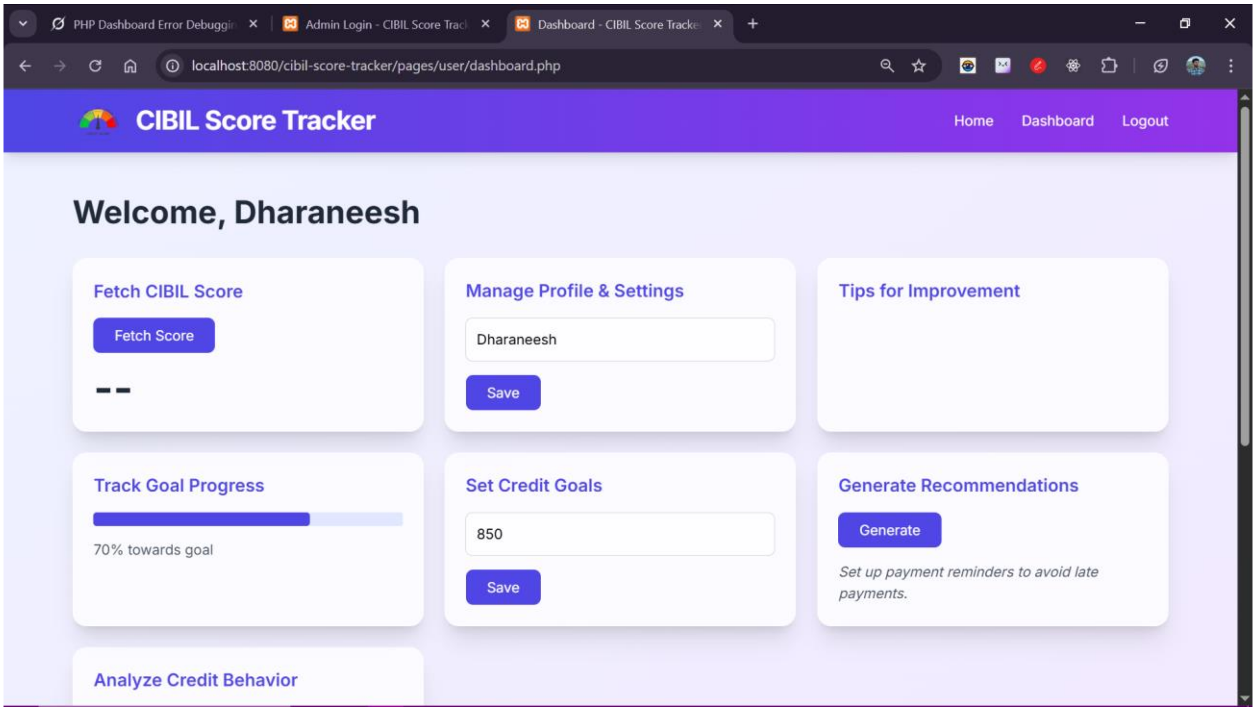

PHP Dashboard Error Debuggin
Admin Login - CIBIL Score Track
Login - CIBIL Score Tracker
localhost:8080/cibil-score-tracker/pages/user/login.php
Login

Sign In

Email: Dharaneesh

Password: ••••••••••

Don't have an account? Register


PHP Dashboard Error Debuggin
Admin Login - CIBIL Score Track
Dashboard - CIBIL Score Tracker
localhost:8080/cibil-score-tracker/pages/user/dashboard.php
CIBIL Score Tracker
Home
Dashboard
Logout
Fetch Score
Save
Save
Generate

Welcome, Dharaneesh

Fetch CIBIL Score

--

Manage Profile & Settings

Dharaneesh

Tips for Improvement

Track Goal Progress

70% towards goal

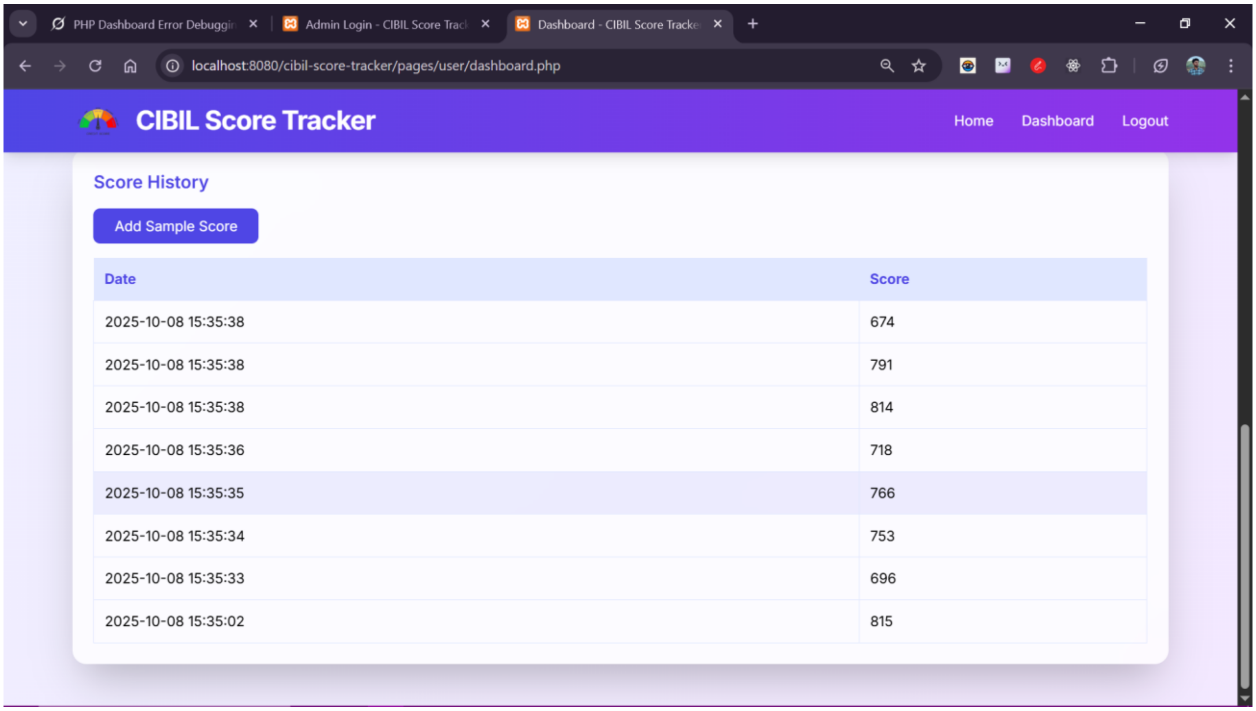
Set Credit Goals

850

Generate Recommendations

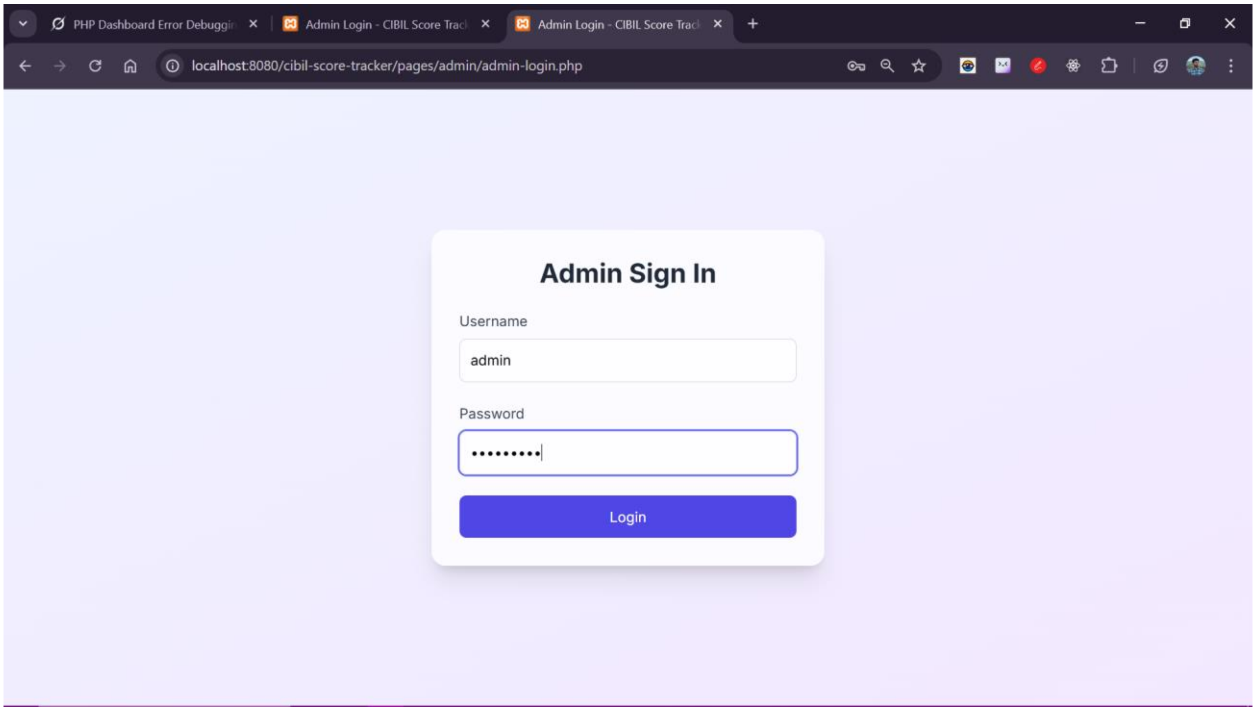
Set up payment reminders to avoid late payments.

Analyze Credit Behavior


PHP Dashboard Error Debuggir
Admin Login - CIBIL Score Trac
Dashboard - CIBIL Score Tracke
localhost:8080/cibil-score-tracker/pages/user/dashboard.php
CIBIL Score Tracker
Home
Dashboard
Logout
Add Sample Score

Score History

Date	Score
2025-10-08 15:35:38	674
2025-10-08 15:35:38	791
2025-10-08 15:35:38	814
2025-10-08 15:35:36	718
2025-10-08 15:35:35	766
2025-10-08 15:35:34	753
2025-10-08 15:35:33	696
2025-10-08 15:35:02	815


PHP Dashboard Error Debuggir
Admin Login - CIBIL Score Trac
Admin Login - CIBIL Score Trac
localhost:8080/cibil-score-tracker/pages/admin/admin-login.php
Login

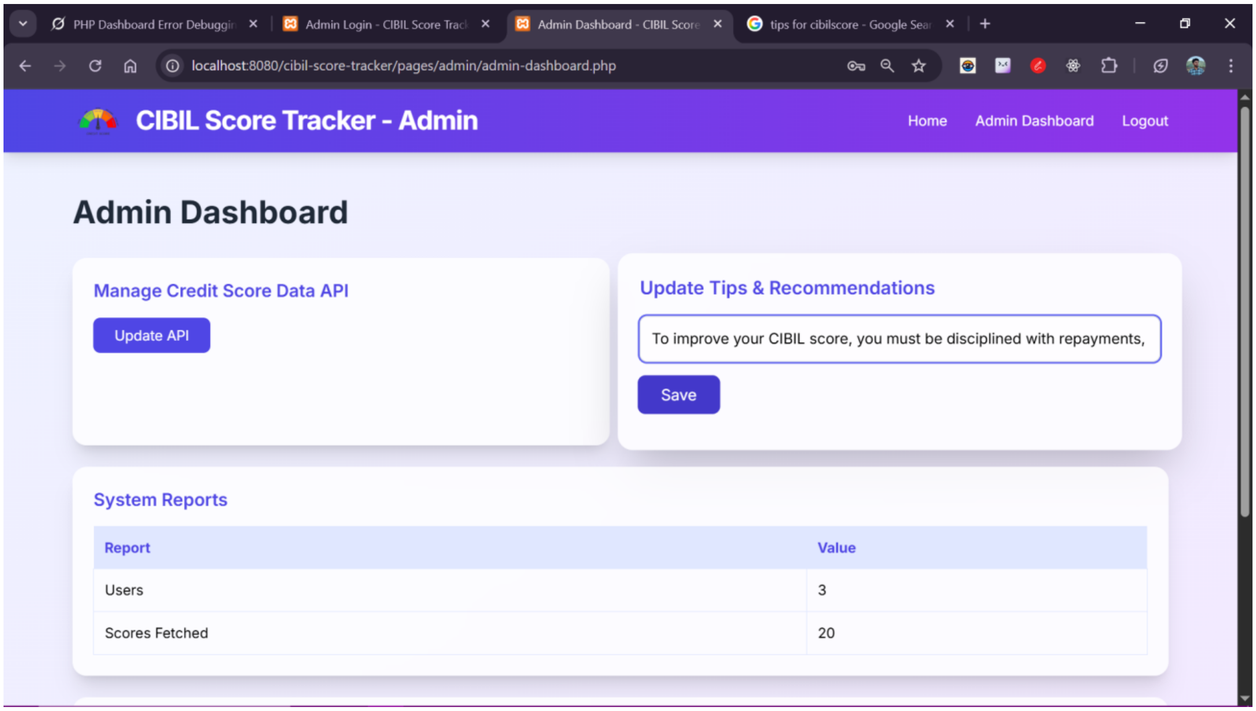
Admin Sign In

Username

admin

Password

•••••••••


PHP Dashboard Error Debuggir
Admin Login - CIBIL Score Track
Admin Dashboard - CIBIL Score
tips for cibilscore - Google Sear
localhost:8080/cibil-score-tracker/pages/admin/admin-dashboard.php
CIBIL Score Tracker - Admin
Home
Admin Dashboard
Logout

Admin Dashboard

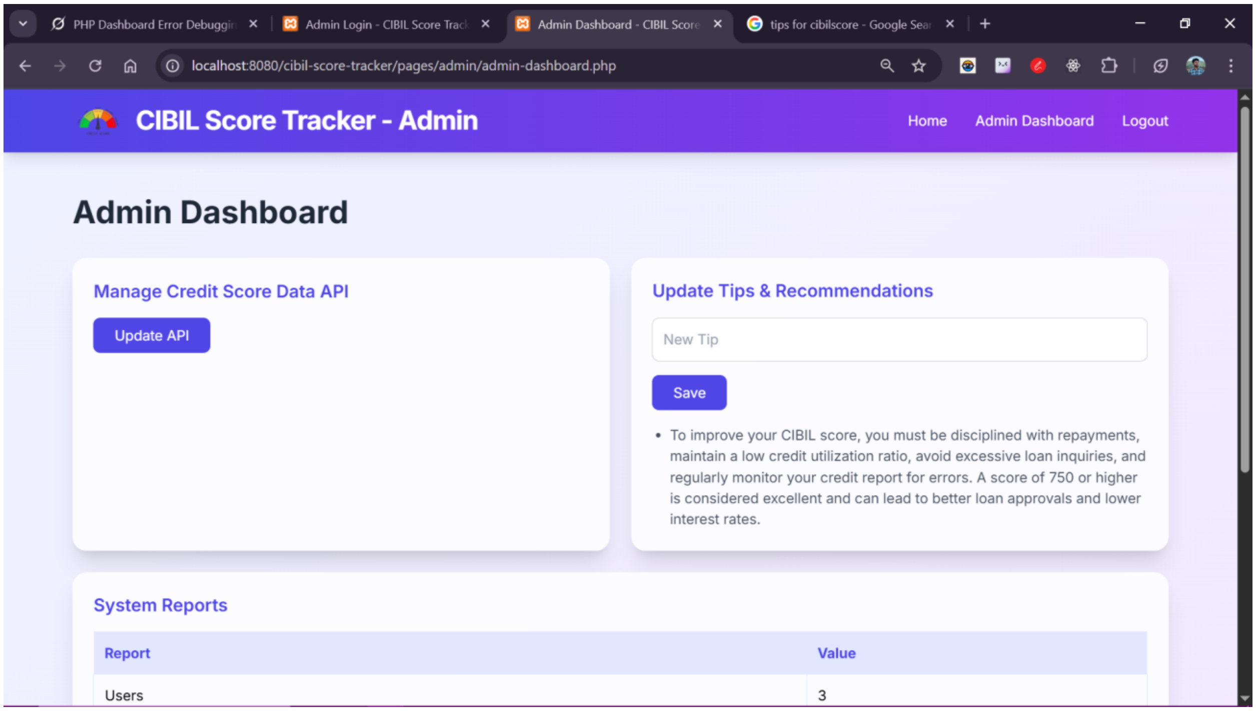
Manage Credit Score Data API

Update Tips & Recommendations

To improve your CIBIL score, you must be disciplined with repayments,

System Reports

Report	Value
Users	3
Scores Fetched	20


PHP Dashboard Error Debuggir
Admin Login - CIBIL Score Track
Admin Dashboard - CIBIL Score
tips for cibilscore - Google Sear
localhost:8080/cibil-score-tracker/pages/admin/admin-dashboard.php
CIBIL Score Tracker - Admin
Home
Admin Dashboard
Logout

Admin Dashboard

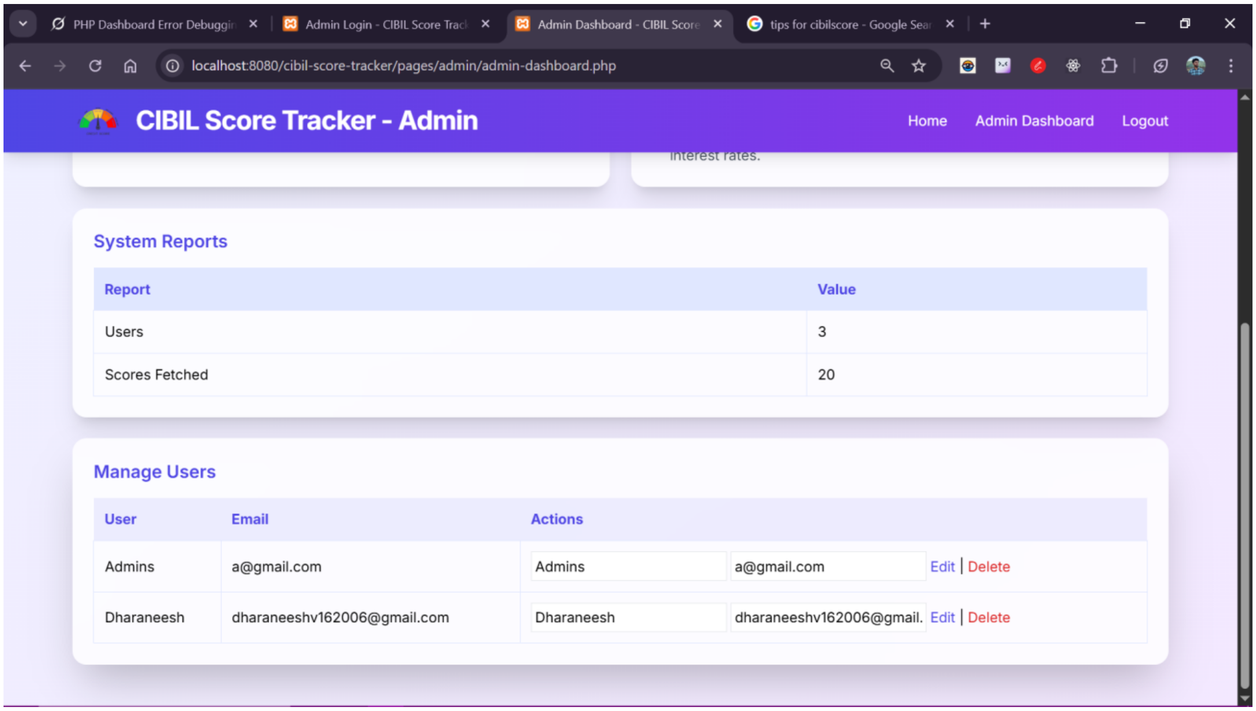
Manage Credit Score Data API

Update Tips & Recommendations

- To improve your CIBIL score, you must be disciplined with repayments, maintain a low credit utilization ratio, avoid excessive loan inquiries, and regularly monitor your credit report for errors. A score of 750 or higher is considered excellent and can lead to better loan approvals and lower interest rates.

System Reports

Report	Value
Users	3


PHP Dashboard Error Debuggin
Admin Login - CIBIL Score Track
Admin Dashboard - CIBIL Score
tips for cibilscore - Google Sear
localhost:8080/cibil-score-tracker/pages/admin/admin-dashboard.php
CIBIL Score Tracker - Admin
Home
Admin Dashboard
Logout
interest rates.

System Reports

Report	Value
Users	3
Scores Fetched	20

Manage Users

User	Email	Actions
Admins	a@gmail.com	Admins a@gmail.com Edit | Delete
Dharaneesh	dharaneeshv162006@gmail.com	Dharaneesh dharaneeshv162006@gmail. Edit | Delete