

Group 9

DebtSetGo – Personal Financer

Software Engineering Project

Names:

Jigyasa Jha
Mohsin Hussain
Zihan Tang
Fikru Gedeon
Prithvi Kurmi

Semester: Fall 2025

Group Number: 9

Coordinator: Zihan Tang

Name of the Guide: Dr. Tushara Sadasivuni

Date: (10/06/2025)

TABLE OF CONTENTS

1.0	<i>INTRODUCTION</i>	3
1.1	<i>Software Engineers' information</i>	3
1.2	<i>Planning and Scheduling</i>	3
1.3	<i>Teamwork Basics</i>	4
1.4	<i>Problem Statement</i>	4
1.5	<i>System Requirements</i>	5
1.5.1	<i>Context Diagram</i>	5
1.5.2	<i>Activity Diagram</i>	6
2.0	<i>REQUIREMENTS</i>	7
2.1	<i>Use Cases</i>	7
2.2	<i>Requirements</i>	7
2.3	<i>Use Case Diagrams</i>	7
3.0	<i>DATABASE TABLES</i>	8
4.0	<i>GITHUB</i>	8
	<i>References</i>	8

1.0 INTRODUCTION

1.1 Software Engineers' information

Jigyasa Jha: Senior computer science and political science double major

Mohsin Hussain:

Zihan Tang: First year CS PhD student, research area is VR, AR, security, privacy

Fikru Gedeon: Senior computer science major

Prithvi Kurmi: Junior Computer Science major, STEM Tutor CS, SI Leader Calc II, Backend Developer

1.2 Planning and Scheduling

Assignee Name	Email Address	Task	Duration (hours)	Dependency	Due Date	Evaluation
Jigyasa Jha	jjha1@student.gsu.edu	3 Use Cases, 3 Requirements, 1 Diagram	3	draw.io	10/06/2025	100%
Mohsin Hussain	mhussain11@student.gsu.edu	3 Use Cases, 3 Requirements, 1 Diagram	2		10/06/2025	100%
Zihan Tang	ztang15@student.gsu.edu	3 Use Cases, 3 Requirements, Database	5	Github Project Create	10/06/2025	100%
Fikru Gedeon	gfikru@student.gsu.edu	3 Use Cases, 3 Requirements, 1 Diagram	2	None	10/06/2025	100%
Prithvi Kurmi	pkurmi@student.gsu.edu	3 Use Cases, 3 Requirements, 1 Diagram	2	Writing	10/06/2025	100%

1.3 Teamwork Basics

Ground rules:

- Each team member should contribute to the project to the best of their abilities.
- Each team member should take part in group meetings, if any team member is unable to take part in a group meeting, they should let the rest of their group members know ahead of time.
- Team members are expected to listen to opinions shared by other team members respectfully and without interrupting.
- Team members are expected to take ownership of assigned tasks and inform the group of any obstacles that arise while working on their assigned task.
- Team members are expected to follow project guidelines.
- When facing difficult behaviors from other team members. The rest of the team will discuss what the source of the problem is and decide how to proceed

When facing group problems with the project, team members can call for a meeting or directly contact any specific team member to resolve the issue.

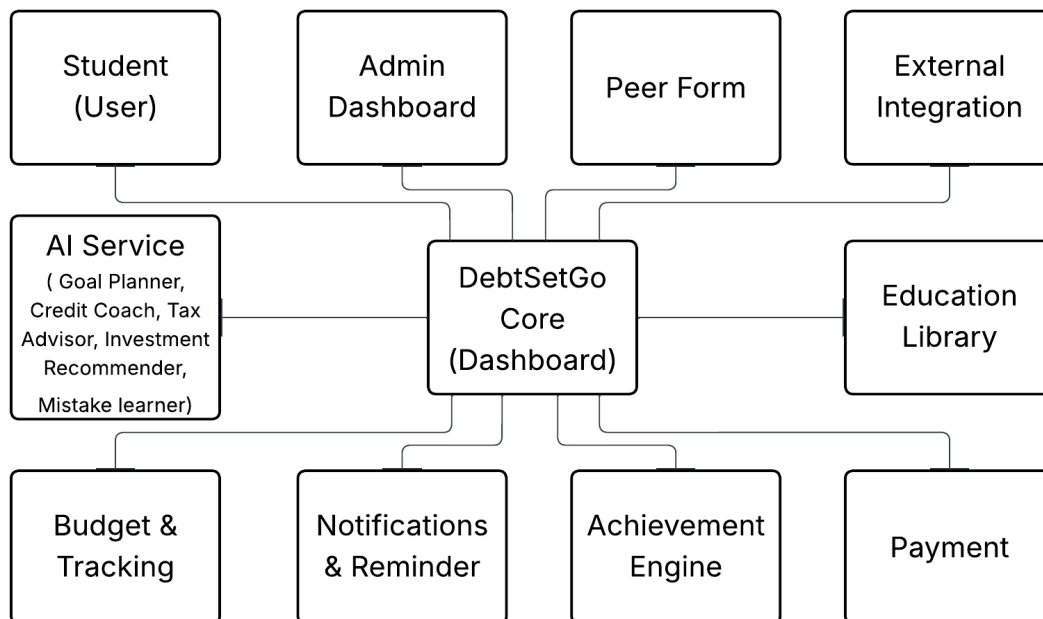
1.4 Problem Statement

- DebtSetGo is an AI-powered personal finance platform designed for students to manage credit, taxes, savings, and investments in one place.
- Features include:
 - **User Management:** Handles login, signup, and profile personalization.
 - **AI Goal Planner:** Generates step-by-step financial roadmaps.
 - **AI Credit Coach:** Provides credit-building tips and reminders.
 - **State-Wise Tax Advisor:** Offers state-specific tax rules and refund estimates.
 - **Investment Recommender:** Suggests safe, student-friendly investment options.
 - **What-If Simulator:** Compares scenarios like saving vs. investing.
 - **AI Mistake-Learner:** Learns user patterns and prevents repeat mistakes.
 - **Budget Tracker:** Tracks income, expenses, and bills.
 - **Goal Dashboard:** Displays progress visually with charts.
 - **Notifications & Reminders:** Sends alerts for bills, deadlines, and goals.
 - **Educational Library:** Provides simple guides on financial topics.
 - **Peer Forum + Admin Dashboard:** Community space with moderation.
 - **Achievement Cards:** Auto-generates social media-friendly success cards.
 - **Smart Payment Suggestions:** Helps maximize credit card rewards.
- **Who is DebtSetGo for?** College students and young adults who are managing finances for the first time.
- **What problem does it solve?** It prevents costly financial mistakes (e.g., bad loans, missed payments) and teaches financial literacy in an actionable, student-friendly way
- **What alternatives are available?** Mint, Credit Karma, NerdWallet...

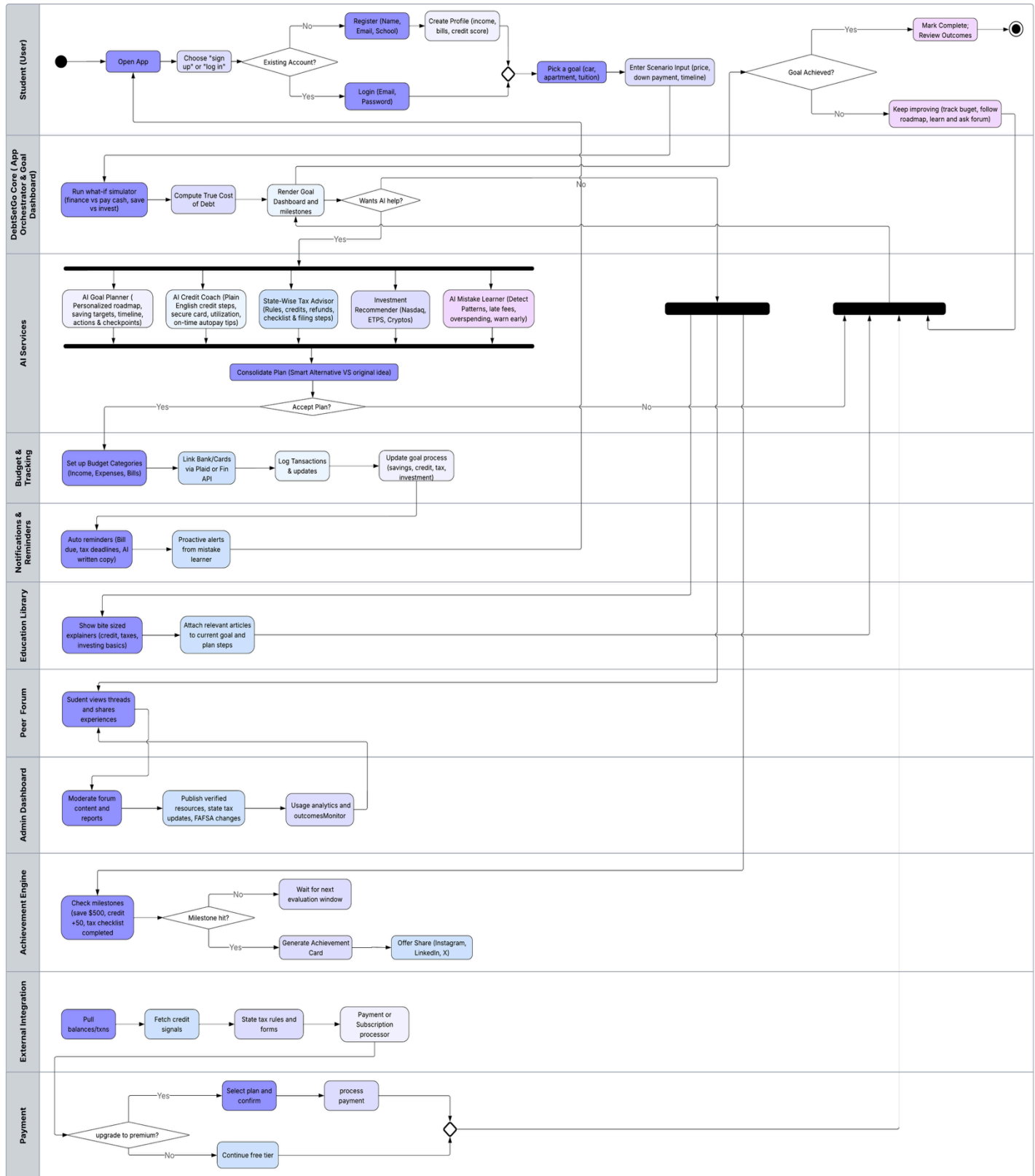
- **Why is this project compelling and worth developing?** It addresses a unique pain point for students who lack personalized financial guidance and combines education, AI-driven insights, and motivation.
- **Top-level objectives & differentiators:**
 - Objectives: Help students build assets, not debt; provide personalized planning; increase financial literacy.
 - Differentiators: Student-first design, AI-driven goal setting, plain-English explanations, social and motivational features.
- **Target Customers:** U.S. college students and young adults (ages 18–25).
- **Scope:** Mobile-friendly web app (future mobile app option) supports login for students and admin dashboard.
- **Competitors and novel approach:**
 - Competitors: Mint, Credit Karma, NerdWallet
 - Novelty: AI-based goal planner, proactive mistake prevention, integrated tax and credit help, student community.
- **Interesting technical points:**
 - AI-driven personalized financial planning
 - Predictive mistake prevention
 - Interactive financial simulations and visualizations
- **Login features:** Yes, it includes Client Login

1.5 System Requirements

1.5.1 Context Diagram



1.5.2 Activity Diagram



2.0 REQUIREMENTS

2.1 Use Cases

Use Case no.: 01

Use Case Name: User Registration and Login

Actors: Student (User)

Description: A new user signs up with email and password, verifies email, and logs in. Returning users authenticate and reach the dashboard.

Alternate Path: Social login (school SSO) if enabled.

Exception Path: Invalid credentials; locked account after N failed attempts; email not verified.

Pre-condition: System reachable; user not authenticated.

Post-condition: Auth session created; user profile stub stored.

Use Case no.: 02

Use Case Name: Profile Personalization

Actors: Student (User)

Description: User sets name, school and state, income sources, credit card ownership, financial goals preference.

Alternate Path: Skip non mandatory fields; import from previous survey.

Exception Path: Invalid state code; network error while saving.

Pre-condition: User authenticated.

Post-condition: Profile updated; personalization saved.

Use Case no.: 03

Use Case Name: Budget Setup and Tracking

Actors: Student (User)

Description: Create monthly budget with categories; system auto categorizes transactions and shows variances.

Alternate Path: Start from template; copy last month's budget.

Exception Path: Category conflicts; invalid amounts.

Pre-condition: Transactions exist or user defined categories.

Post-condition: Budget and budget items saved, progress visible.

Use Case 04 – AI Goal Planner

Actors: Student (User)

Description: User provides financial goals such as saving, loan repayment, or building credit. The system's AI Goal Planner analyzes income and expenses to generate a personalized saving plan with milestones.

Alternate Path: User imports a goal template or links previous budget data.

Exception Path: Invalid inputs or missing income data.

Pre-Condition: User authenticated and profile exists.

Post-Condition: Goal data and monthly targets stored in database.

Use Case 05 – Budget Analytics & Visualization

Actors: Student (User)

Description: The system retrieves budget and transaction data, categorizes expenses, and displays charts showing planned vs. actual spending. Alerts are generated for overspending.

Alternate Path: User changes time frame or export report to CSV.

Exception Path: Missing transaction data or database error.

Pre-Condition: Budget data exists.

Post-Condition: Graphs and alerts visible on dashboard.

Use Case 06 – AI Mistake Learner

Actors: Student (User)

Description: System analyzes past financial behaviors and detects repeat mistakes such as late payments. It then sends preventive tips to avoid recurrence.

Alternate Path: User snoozes alerts for specific patterns.

Exception Path: Insufficient historical data to train AI model.

Pre-Condition: Minimum three months of activity available.

Post-Condition: Recommendations recorded and displayed under “Insights.”

Use Case 07 – Financial Progress Report Generator

Actors: Student (User)

Description: The system compiles a detailed monthly report showing income, expenses, savings progress, and goal achievements. It highlights improvements and provides personalized suggestions for better money management.

Alternate Path: User selects a custom date range or exports the report as a PDF/CSV.

Exception Path: Missing data for selected period or report generation timeout.

Pre-Condition: User must have at least one month of recorded transactions and goals.

Post-Condition: Report generated, saved in user history, and available for download.

Use Case 08 – AI Financial Advisor Chatbot

Actors: Student (User)

Description: An AI-powered chatbot provides real-time financial advice, answers questions about budgeting, goals, and spending habits, and suggests smart actions based on user data.

Alternate Path: User asks generic financial literacy questions or requests detailed trend analysis.

Exception Path: AI service unavailable or insufficient user data for personalized responses.

Pre-Condition: User authenticated; chatbot feature enabled.

Post-Condition: Advice and recommendations displayed in chat; session logged for analytics.

Use Case 09 – Investment Tracker and Advisor

Actors: Student (User)

Description: The system allows users to log their small investments (e.g., savings accounts, ETFs, or crypto). It tracks performance, calculates gains/losses, and provides AI-based suggestions for diversification or risk reduction.

Alternate Path: User imports investment data automatically from a connected bank or portfolio app.

Exception Path: Invalid investment type, incorrect balance data, or failed third-party API connection.

Pre-Condition: User authenticated and investment tracking feature enabled.

Post-Condition: Investment portfolio updated and analytics shown in the “Wealth Insights” section.

Use Case 09 – AI Credit Coach

Actors: Student (User)

Description: The system provides personalized credit-building advice, detects factors lowering credit scores, and suggests actions like paying bills on time or reducing utilization.

Alternate Path: User imports credit report or data manually.

Exception Path: User don't have access to credit report

Pre-Condition: credit data available.

Post-Condition: Credit score insights and improvement tips displayed on dashboard.

Use Case 10 – State-Wise Tax Advisor

Actors: Student (User)

Description: The system provides state-specific tax guidance, refund estimations, and filing deadlines based on the user's location and income.

Alternate Path: User manually inputs state or uploads tax document.

Exception Path: Invalid or unsupported state code; missing income data.

Pre-Condition: User profile completed with state info.

Post-Condition: Personalized tax summary and recommendations displayed.

Use Case 11 – Educational Library & Quiz Module

Actors: Student (User)

Description: Users access short articles, videos, and quizzes on financial literacy topics (budgeting, credit, investing).

Alternate Path: User downloads resources or saves for later.

Exception Path: Missing resource files or connection timeout.

Pre-Condition: User authenticated.

Post-Condition: Learning progress saved; quiz scores recorded.

Use Case 12 – Family / Guardian Access

Actors: Student (User), Guardian (Parent/Advisor)

Description: A student can invite a guardian with limited view or manage permissions so the guardian can monitor budgets, goals, and bills (useful for dependent students or financial coaches).

Alternate Path: Guardian access via one-time access code or school-authorized guardian connection.

Exception Path: Guardian account not verified; guardian tries actions beyond granted permissions.

Pre-condition: Student authenticated and has completed profile; guardian account exists or invited.

Post-condition: Guardian linked with specified permissions; audit log entry recorded.

Use Case 13 – Bank/Account Linking & Data Import

Actors: Student (User), External Financial Institution (via API/CSV)

Description: User connects bank(s) or credit accounts (via secure aggregator or manual CSV) to import transaction history and account balances for automatic categorization and reconciliation.

Alternate Path: Manual CSV upload or manual transaction entry if API linking is unavailable.

Exception Path: Failed third-party API auth; stale or inconsistent CSV formats.

Pre-condition: User authenticated and consents to data linking.

Post-condition: Transactions imported, matched, and available for categorization and budgeting.

Use Case 14 – Fraud Detection & Dispute Workflow

Actors: Student (User), Fraud Team / Bank (System)

Description: System flags unusual transactions (high amount, new payee, rapid frequency) and allows user to review/confirm or start a dispute workflow that notifies the linked bank or internal fraud team.

Alternate Path: User marks transaction as safe; system learns to reduce false positives.

Exception Path: Insufficient transaction metadata to evaluate; user unreachable for confirmation.

Pre-condition: Connected accounts and transaction data present.

Post-condition: Transaction flagged status saved; dispute ticket created (if initiated) and user notified.

Use Case 15 – Admin Dashboard & System Monitoring

Actors: Admin, Support Agent

Description: Admin can view system health, user activity metrics, manage content (educational resources), handle user support tickets, and perform escalations.

Alternate Path: Read-only analytics role for product/product management; escalate to engineering for incidents.

Exception Path: Admin action attempted without sufficient privileges.

Pre-condition: Admin authenticated with RBAC role.

Post-condition: Admin actions recorded in audit logs; support ticket state updated.

2.2 Requirements

Requirement number: 01

Use Case number: 01

Introduction: Secure account creation and login.

Inputs: Email, password, optional SSO token.

Requirements Description: Must hash passwords; Email verification before full access; Rate-limit login attempts; lockout after N tries.

Outputs: Auth session; verification email sent.

Requirement number: 02

Use Case number: 02

Introduction: Store and update user profile data.

Inputs: name, state_code, school, preferences.

Requirements Description: Validate state_code against allowed list; Partial updates supported; Audit profile changes.

Outputs: Updated profile record.

Requirement number: 03

Use Case number: 03

Introduction: Budget creation and tracking.

Inputs: month, categories, planned amounts.

Requirements Description: Auto categorize transactions with user overrides; Compute planned vs. actual per category; Support copy from previous month.

Outputs: Budget, budget_items, monthly variance.

Requirement 04 – AI Goal Planner

Input: Goal type, target amount, time frame, income.

Process: AI algorithm computes monthly savings plan and expected completion timeline.

Output: Goal plan and visual progress chart.

Constraints: Goal data must auto-update monthly; secure storage required.

Requirement 05 – Budget Analytics

Input: User's transactions and budgets.

Process: Auto-categorize expenses, compute variance, generate graphs.

Output: Pie charts, trend lines, and overspending alerts.

Constraints: Charts refresh daily; data stored in MySQL tables.

Requirement 06 – AI Mistake Learner

Input: User transaction history and payment logs.

Process: Identify repetitive mistakes and recommend preventive steps.

Output: Personalized alerts and reminder notifications.

Constraints: AI model must retrain weekly as new data arrives.

Requirement 07 – Expense Forecasting (AI Predictor)

Input: Historical spending data, income, and seasonality factors.

Process: AI model predicts next-month expenses per category using trend analysis and user habits.

Output: Predicted expense report and deviation risk levels.

Constraints: Forecasts update automatically on the 1st of each month; model retrained quarterly using aggregated anonymized data.

Requirement 08 – Collaborative Budget Sharing

Input: Budget ID, list of invited users (emails), access permissions (view/edit).

Process: Allow shared household or team budgets where multiple verified users can collaborate.

Output: Shared budget dashboard, change logs, and notifications on edits.

Constraints: Each change must be attributed to a user; sharing only allowed with verified accounts; audit trail maintained for 90 days.

Requirement 09 – Smart Bill Reminder System

Input: Bill name, due date, amount, linked payment method.

Process: Schedule reminders, detect recurring bills automatically from transactions, and notify users before due dates.

Output: Push/email reminders and a visual calendar view.

Constraints: Minimum two reminders per bill (3 days before and same day); store reminder logs securely; integrate with AI Mistake Learner for missed payment patterns.

Requirement 09 – AI Credit Coach

Input: User's credit score, payment history, and credit utilization.

Process: AI analyzes patterns, generates personalized credit-improvement tips.

Output: Credit tips and milestone tracker.

Constraints: Credit data must refresh monthly; comply with privacy laws (e.g., FCRA)

Requirement 10 – State-Wise Tax Advisor

Input: User income, location, deductions.

Process: Retrieve state tax data, calculate estimated tax/refund.

Output: Tax summary and filing suggestions.

Constraints: Must update tax rules annually; support all U.S. states.

Requirement 11 – Educational Library & Quiz Module

Input: Topic selection or keyword search.

Process: Retrieve content, track user progress, compute quiz results.

Output: Progress badge and learning report.

Constraints: Content must be mobile-friendly; store user activity logs for insights.

Requirement 12 – Secure Bank Integration & Data Import

Use Case number: 13

Introduction: Importing transaction/account data from external institutions.

Inputs: OAuth token / secure aggregator token or CSV file upload.

Requirements Description: Use industry-standard aggregators (or OAuth) for account linking; validate CSV schema and provide mapping UI; token refresh handling; store only needed metadata and encrypted tokens.

Outputs: Imported transactions and account balances mapped to user accounts.

Constraints: Must follow PCI-esque best practices for handling financial tokens; retry/backoff for API failures; user must be able to revoke access.

Requirement 13 – Notification & Communication System

Use Case number: (applies to multiple use cases: 05, 06, 09, 14)

Introduction: Deliver push, email, and SMS messages for reminders, alerts, and marketing (opt-in).

Inputs: Notification trigger (due date, overspend, fraud flag), user contact preferences.

Process: Queue message, apply user preferences and throttling, send via provider, log delivery and failures.

Output: Notification record, delivery receipt, retry logs.

Constraints: Rate-limit outbound messages per user; comply with CAN-SPAM, TCPA and local privacy laws; support opt-out and preference center.

Requirement 14 – Fraud Detection & Dispute Handling

Use Case number: 14

Introduction: Detect suspicious activity and enable dispute resolution.

Inputs: Transaction feed, user behavioral baselines, device/location signals.

Process: Rule-based + ML scoring to flag anomalies, prompt user verification, allow dispute ticket creation routed to bank or support.

Output: Flagged transaction status, dispute ticket, resolution timeline.

Constraints: Flagged incidents must be handled within SLA (e.g., initial response within 24 hours); store related logs for at least 12 months; protect PII and comply with applicable financial regulations.

Requirement 15 – Admin Controls, RBAC & Audit Logging

Use Case number: 15

Introduction: Admin capability to manage platform safely and transparently.

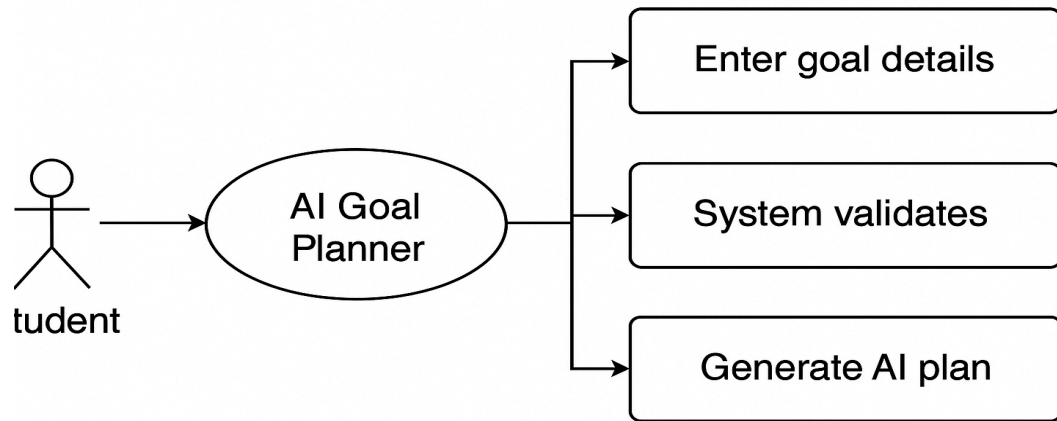
Inputs: Admin actions (user suspension, content edits, report generation).

Process: Enforce role-based access control (RBAC), require MFA for sensitive operations, and write every admin action to immutable audit logs.

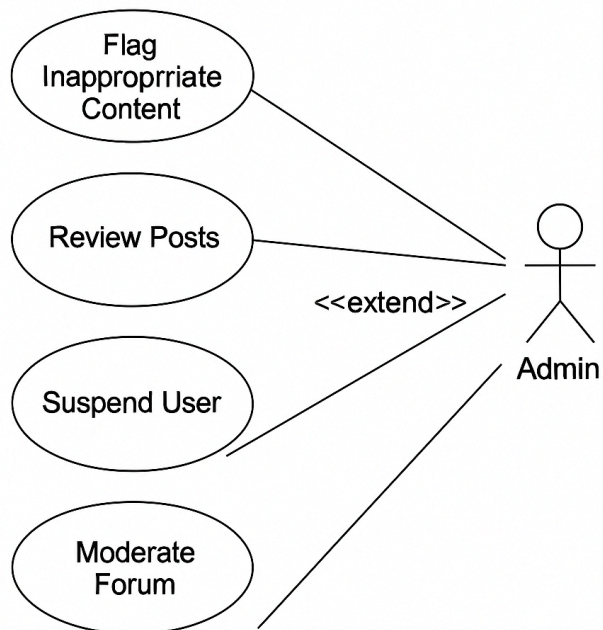
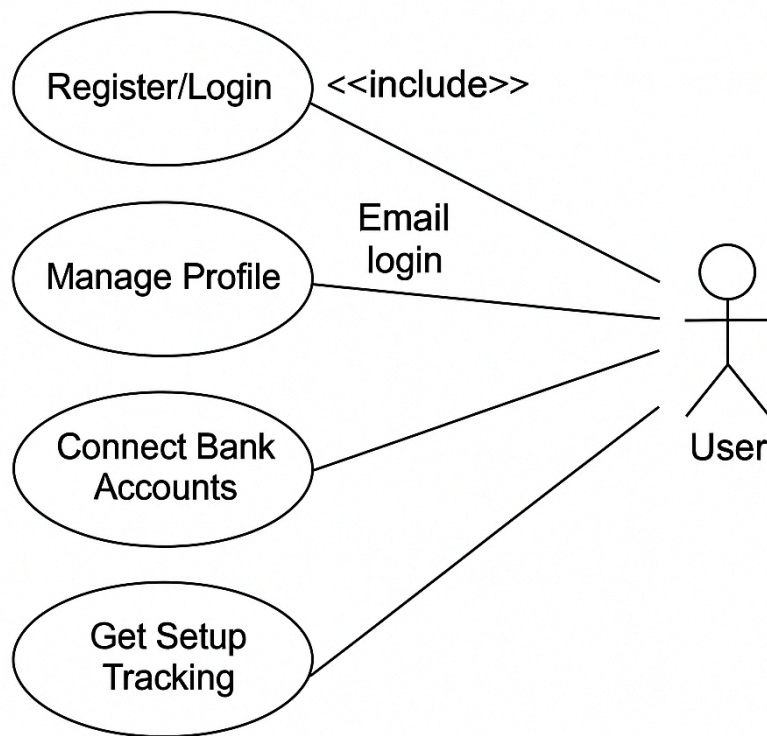
Output: Admin action logs, activity reports, and role change records.

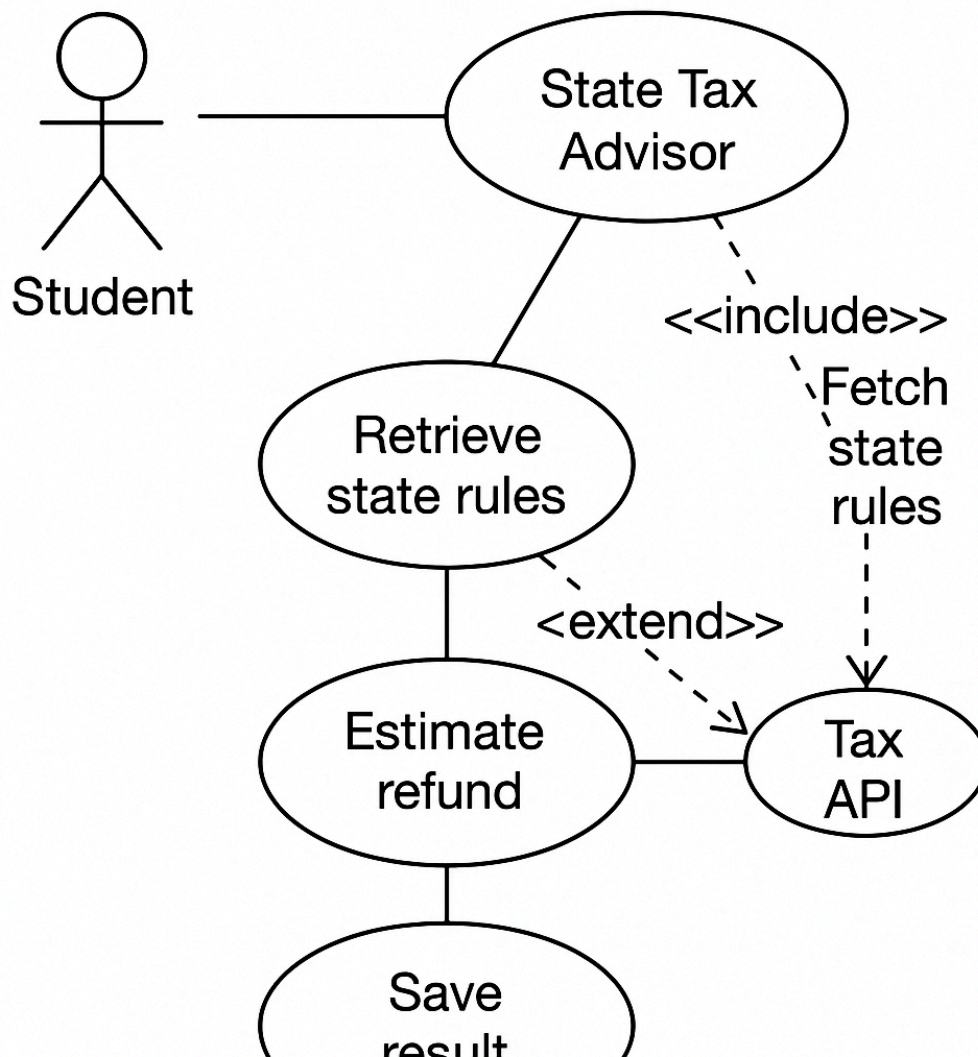
Constraints: Audit logs retained per policy (e.g., 90–365 days); only super-admins can change RBAC definitions; admin UI must mask PII unless explicitly authorized.

2.3 Use Case Diagrams

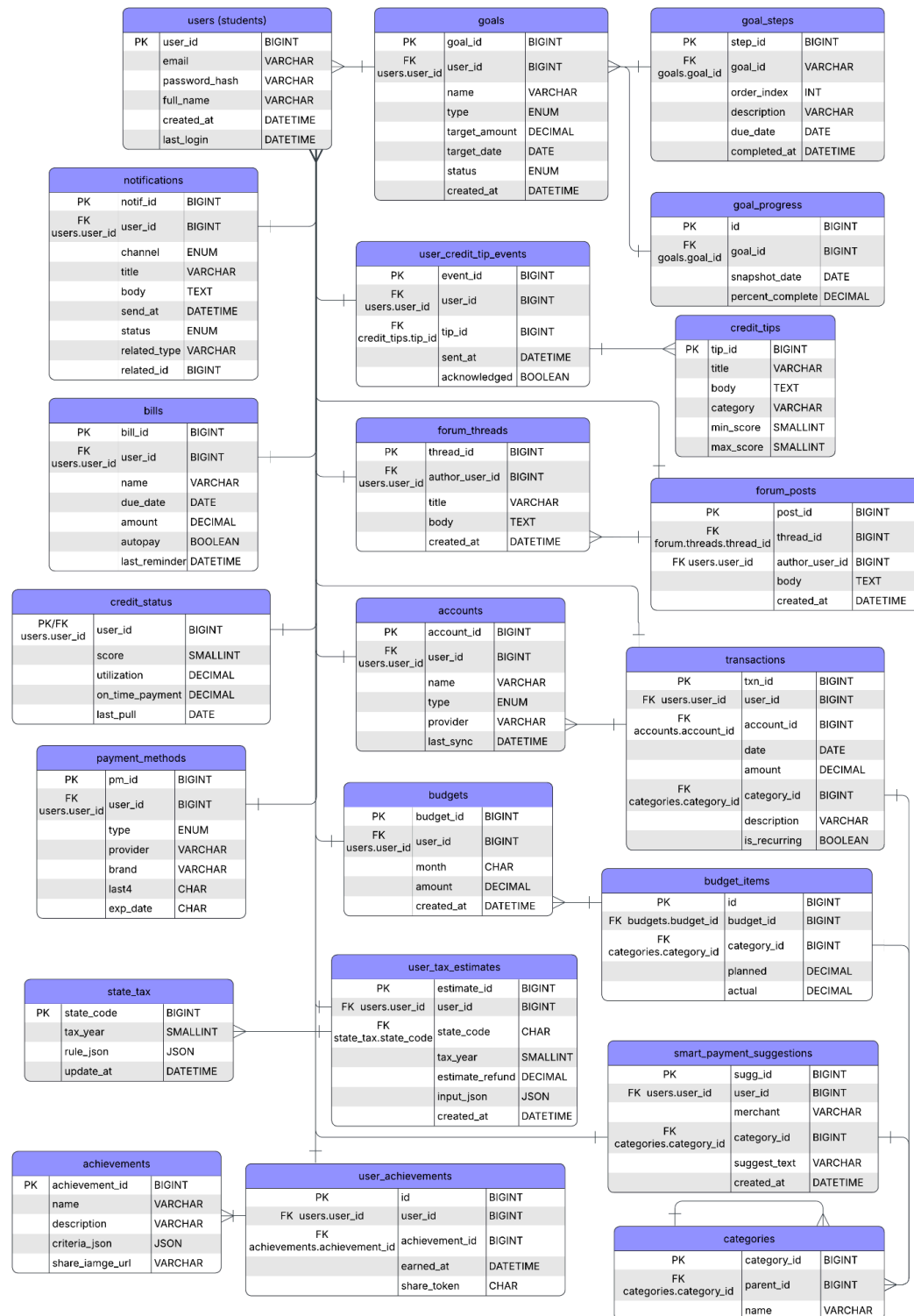


AI Goal Planner



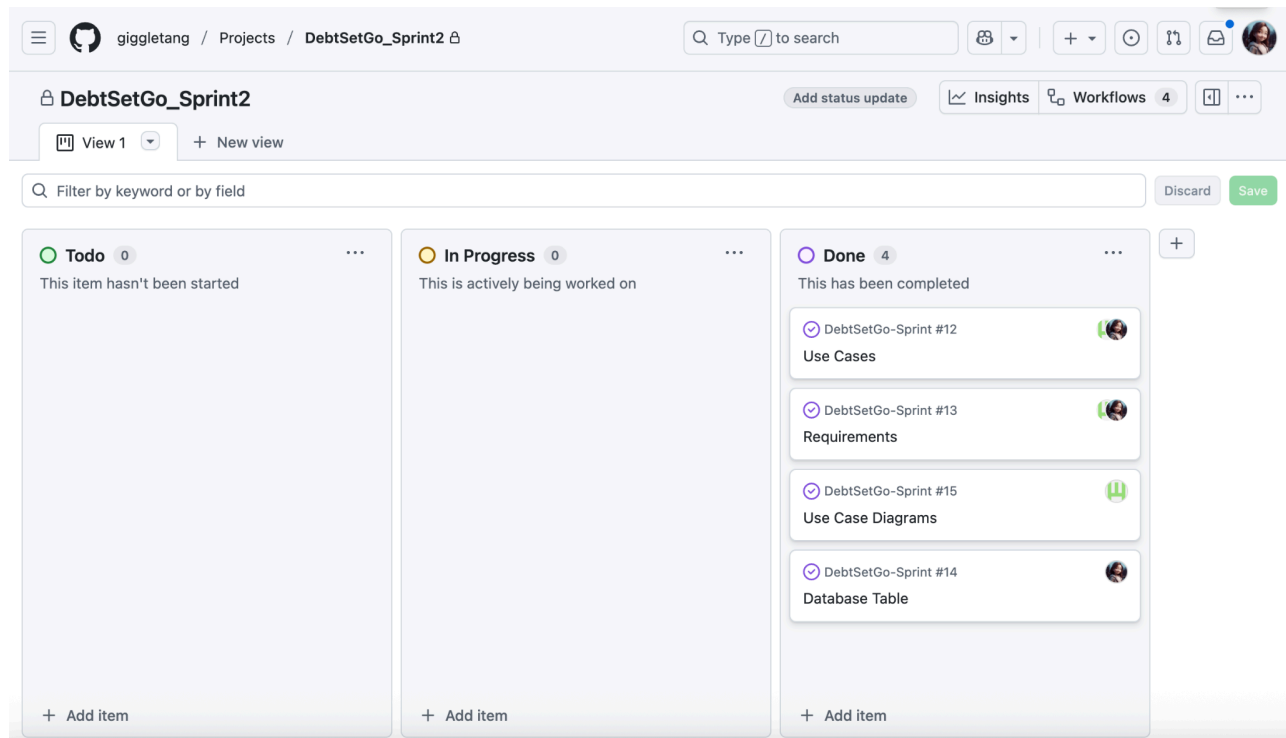


3.0 DATABASE TABLES



We will use MySQL in our projects

4.0 GITHUB



References

- Bhatia, P., & Bhatia, R. (2021). Artificial intelligence in personal finance: Opportunities and challenges. *International Journal of Computer Applications*, 183(29), 12-18. <https://doi.org/10.5120/ijca2021921621>
- National Endowment for Financial Education (NEFE). (2023). Financial Education Resources for College Students. Retrieved from <https://www.nefe.org>
- Mint by Intuit. (n.d.). Budget Tracker & Planner. Retrieved September 10, 2025, from <https://mint.intuit.com>