1. Team info

Provide a concise summary of the project team and project artifacts. Specifically:

- List each team member and their role in the project.

America Pacheco: Front-End Developer (HTML, CSS, UI design)
Bailey Bounnam: Back-End Developer (Node.js, Express, API integration)
Ross Henderson: Team Lead and GitHub Manager (coordinate tasks, oversee GitHub repository)
Brian Fang: Database Manager (MySQL design, secure storage of financial data)
Bryan Partida: API Specialist (Plaid API integration, financial data processing)

- Link to each project relevant artifact such as your git repo (this can be empty for now).

https://github.com/henderos/CS362-Class-Project

- List communication channels/tools and establish the rules for communication.

Our main form of communication will be Discord, where we will establish clear threads for tasks, announcements, and questions. We will try to respond within 17 hours of a message of discord. If no one responds, then politely give a reminder.

2. Product description

- Start with a short and catchy project title (not “CS 362 project pitch”) and your full names.

Personal Finance Analyzer

- Abstract: The first paragraph of your document must be an abstract (or executive summary, or TL;DR) that explains your project at a high level. If someone read nothing but that one paragraph, what do you want them to know?

Personal Finance Analyzer is a web application designed to simplify financial management for users, providing tools to track spending and manage subscriptions. With an intuitive user interface and analytics, it aims to empower users to make informed financial decisions effortlessly.

- Goal: What are you trying to do? For example, what task or problem will your system help users with?

This project will help users to manage and keep track of their personal finances, see where their money is going, and make decisions for future spending based on these facts.

- Current practice: How is it done today, and what are the limits of current practice?

Today, people commonly use spreadsheets, individual banking apps, or third-party apps to track their finances. These methods are often not automated, or if you have accounts in different banks, then you have to individually log into each banks' account. With our web app, you will be able to link every account so that you can see everything in one spot with advanced features such as subscription tracking or personalized savings advice.

- Novelty: What is new in your approach and why do you think it will be more successful than other approaches? Do not reinvent the wheel or reimplement something that already exists, unless your approach is different.

Our finance manager will not only allow users to see their finances, but also will tell users where certain "problem areas" may be, allowing them to see where they are likely able to improve.

- Effects: Who cares? If you are successful, what difference will it make?

The people who will be using our web app are people who want to see all their financial information with a user-friendly interface. If we are successful in our app, our app will show more transparency with how our users' money is being handled. This will help reduce unnecessary expenses by identifying underused subscriptions and help users allocate their income effectively by highlighting spending patterns.

- Technical approach: Briefly describe your proposed technical approach. This may include what system architecture, technologies, and tools you may use.

Our project will employ a scalable and efficient system architecture. The front end will be built using HTML and CSS, with JavaScript to enhance interactivity and responsiveness. For the back end, we will use Node.js with Express to handle server-side functionality and API routing.
Data storage will be managed using MySQL to securely handle user information, while the Plaid API will enable integration with users' bank accounts for retrieving financial data.

- Risks: What is the single most serious challenge or risk you foresee with developing your project on time? How will you minimize or mitigate the risk? Don't state generic risks that would be equally applicable to any project, like "we might run out of time".

One of the most significant challenges we face is ensuring effective and efficient communication within a large team. Miscommunication or delays in responding to inquiries could lead to misunderstandings, duplicated efforts, or even missed deadlines

in some cases. Additionally, with multiple members handling interconnected tasks, it can be challenging to maintain a clear and cohesive understanding of project progress and priorities.

To mitigate this risk, we will establish clear communication expectations from the beginning. As mentioned above, team members will be required to regularly check Discord, our primary communication platform, and respond to messages within 17 hours. Tasks will be clearly assigned and tracked using GitHub Projects to provide transparency and accountability. Regular updates will be shared in designated channels to keep everyone informed of ongoing progress.

Additionally, add the following:

- 4+ major features you will implement.

- **Dashboard Overview:** Display consolidated financial data, including recent transactions, and monthly spending summaries.
- **Subscription Tracking**: Identify and track recurring subscriptions, flagging underused services.
- **Spending Categories:** Provide visual breakdowns of expenses by category (e.g., groceries, rent, entertainment).
- **Budgeting Tool:** Allow users to set budgets for categories and receive alerts when nearing limits.

2+ stretch goals you hope to implement.

- **Implementing AI suggestions and advice for users.**

- **Mobile support.**

For this assignment you should revise the living document for your project and make sure that it contains the following sections.

Start with a short and catchy project title (not "CS 362 project idea") and your full names.
Abstract: The first paragraph of your document must be an abstract (or executive summary, or TL;DR) that explains your project at a high level. If someone read nothing but that one paragraph, what do you want them to know?
Goal: What are you trying to do? for example, what task or problem will your system help users with?
Current practice: How is it done today, and what are the limits of current practice?

Novelty: What is new in your approach and why do you think it will be more successful than other approaches? Do not reinvent the wheel or reimplement something that already exists, unless your approach is different.
Effects: Who cares? If you are successful, what difference will it make?
Use Cases (Functional Requirements): Each team member must come up with and describe at least one use case of your product. Refer to "1. Use Cases (Functional Requirements) " in Project Requirements Elicitation
Non-functional Requirements: Describe at least three non-functional requirements, e.g., related to scalability, usability, security and privacy, etc.
External Requirements: Refer to "3. External Requirements" in Project Requirements Elicitation.
Technical approach: Briefly describe your proposed technical approach. This may include what system architecture, technologies, and tools you may use.
Risks: What is the single most serious challenge or risk you foresee with developing your project on time? How will you minimize or mitigate the risk? Don’t state generic risks that would be equally applicable to any project, like “we might run out of time”.
Team info: This should clearly describe each group members' roles and responsibilities.
Timeline: Tentative high-level weekly goals for your project for the coming 9 weeks.