``CS3337 Project Proposal

Group 4: Adam Dixon, Fernando Serrano Perez, Geovanny Montano, Jacob Hertz, Nicholas Trigueros, Ryan Torrez, Steven Partida, Washika Afrozi.

Proposal #1: Visual Data Sifting App

Situation(Adam): Data is an important way to store information, and it is everywhere. Companies and businesses in particular collect data from users constantly. For the average viewer, the sheer amount of data can be overwhelming. With such a large and constant influx of data, there must be an efficient way to sort it and visualize it, right?

Target(Adam): The target audience are those of us who are fascinated with following the growth of a wide variety of subjects including, sports, net operating revenues, social media influencers, or even healthcare statistics

Option 1: Web Application (Steven)

Build a web application where users can log in or create an account. Through the use of Kaggle's public API, users can search or import datasets to visualize or analyze. These datasets along with the user's account information will be stored in our SQL database. On this web application users will be able to store favorite datasets and search history for quicker access.

Pros:

- No download required
- Easily accessible through mobile and desktop with a single URL.
- Easier and faster to update with newer iterations.
- More screen space

Cons:

- Dependent on internet connection
- Dependent on third-party service
- Data privacy

Option 2 (Ryan)**:** We could make a web scraper that collects data and formats a visualizer based on the info collected. Pro: The reasons for making a web scraper include extracting html code for a database and taking a website's data for repurposing to another location. It is also very versatile meaning we could include many different features into the website including a video from the information collected. Con: The possible reason it might not be doable for this project is because

a web scraper takes a lot of trial and error. It could possibly take a while to accomplish something like this based on the complexity of the website being scraped.

Proposal(Ryan): The best option would be to use established datasets to create visualizers with due to the time constraint and how we would have to create a script for the scraper that could take most of the time that we had limiting us on the formatting and user experience when visiting our website. We will build a web application where companies, businesses, and other institutions can opt in to share their data to be compiled for an easy viewing experience.

Product Name(Adam): Visulyfe

Platform(Jacob): Mobile App, Website, Desktop Application, etc

Database Content: (Steven)

- User data
    - Account information
    - Favorites
    - Search history
- Interface
    - Kaggle API to grab datasets

System Integrations(Jacob): (what external systems, what data to send/get)

External Systems: user login authentication (like DUO Mobile, or just email verification)

Data to Send/Get: feedback on app usability, accessibility, and problems, and updates that address such feedback

Proposal #2: Multiplayer Game (Geo)

Situation (Geo): A group of friends wants to play a simple multiplayer game together online. They enjoy strategic games and want a seamless experience where they can track their progress and compete with each other.

Target(Geo): The customer objective is to create a multiplayer game that provides a fun and engaging experience for friends to play together, while also allowing them to track their progress and compete on leaderboards.

Option 1: Solution: Develop a turn-based strategy game where players take turns making moves on a virtual game board. Implement a backend database to store player profiles, game states, and leaderboard information. Pro: The turn-based nature of the game allows players to strategize and plan their moves carefully, adding depth to the gameplay experience. Con: Turn-based games may require longer wait times between player actions, potentially reducing the pace of the game and causing some players to lose interest.

Option 2: Solution: Create a real-time multiplayer game where players engage in fast-paced battles or challenges against each other. Utilize a backend database to store player profiles, game session data, and leaderboard information. Pro: Real-time gameplay adds excitement and intensity to the gaming experience, keeping players engaged and entertained. Con: Developing real-time multiplayer functionality can be more complex and resource-intensive, requiring robust server infrastructure and potentially leading to technical issues such as lag or synchronization problems.

Proposal(Geo): - We propose Option 1: developing a turn-based strategy game. While real-time gameplay may offer excitement, the turn-based approach allows for more thoughtful decision-making and accommodates players who prefer a slower pace. Additionally, the simplicity of turn-based mechanics can make the game more accessible to a wider audience. With a backend database to manage player profiles and game data, we can still provide features like leaderboards to enhance the competitive aspect of the game.

Product name: **Washika :**

The Quest!

Platform: Executable Application **Washika**
Our Platform would be A website, Mobile App, Desktop Application, etc

Database Content: Fernando

- Player profiles (username, statistics, achievements)
- Game states (current board configuration, player turns, game progress)
- Leaderboard information (player rankings, win/loss records)

System Integrations: Fernando

- External systems: Authentication service for user login
- Data to send/get: Player profile updates, game state synchronization, leaderboard data retrieval