Team 39 Product Backlog
Enjoyagoals

Team Members:
Prahas Pattem, Nick Norton, Karim Mammadli, Nabi Nabiyev, Ryan Chang

Problem Statement:

As humans, we have made almost every facet of life as enjoyable as possible—except for organizing and working towards goals. Well-defined goals and planning have always been at the foundation of society. As society has progressed, we have developed ways to make structuring projects *easier* but not necessarily more fun. Why not? Nearly everything in people's lives requires some amount of planning. Although there are existing services that allow you to organize your goals, they fail to take advantage of people's desire to receive instant gratification and to have fun. Enjoyagoals is unique because it is based on the idea that people like games and reward systems. This application motivates people to be productive because it makes tasks or projects feel like a game, capitalizing on our desire for immediate rewards, pleasure, and feedback.

Background Information:

Audience

The target audience is broad, covering almost anyone who wishes to collaborate with anyone else, or even people who simply want a better way to organize their to-dos. Despite this, very few existing options even attempt to make their product anything beyond a means to an end, failing to utilize our desire for instant gratification.

Similar Platforms

There are several existing goal organization services such as Zenhub, Trello, and Jira. These services allow for easy bug tracking and project management by listing out tasks to be completed, tasks in progress, and finished tasks. Trello focuses on flexibility and ease of use, Jira offers more comprehensive bug tracking features and project tracking at the cost of increasing complexity, and Zenhub integrates the above mentioned features into Github.

Limitations

These existing platforms are very useful, comprehensive, and have intuitive interfaces. However, they are focused entirely on being a tool to help people manage productivity. They could utilize people's desire for entertainment and fun to motivate them to make progress, and thus not only help manage productivity but directly incentivize it in a way that is enjoyable. By focusing on making an application that is fun in addition to being useful, there is a lot more potential for applications that achieve the goal of providing a platform that helps users organize the work that needs to be done to complete a project. Will users be positively incentivized to make progress through a fun leveling system? Will they have negative incentives such as losing health points when they fall behind on deadlines? Our goal is to create a platform that can

function similarly to the aforementioned services but also taps into an entire new dimension that others do not, putting focus on motivating the user(s) to be more productive rather than being just another tool to use.

Functional Requirements:
- As an individual user, I would like to be able to:
    1. register for an Enjoyagoals account.
    2. login and manage my Enjoyagoals account, and reset my password.
    3. register using my google account if I do not have an Enjoyagoals account.
    4. easily set a custom profile picture and customize my profile page's theme and banner.
    5. display my email account and phone number on my Enjoyagoals account alongside other personal information for others to see.
    6. view all of the tasks I need to finish in a comprehensive list after logging in
    7. easily check the details of each individual task (description, date created, deadline, point value).
    8. easily view all of my completed tasks in a list along with my total points accumulated and number of tasks completed in a graphic visualization.
    9. join, switch between, or withdraw from different project rooms at any time after signing in.
    10. purchase exclusive profile pictures or emotes with points that I have earned from tasks to modify my user experience.
    11. view a calendar with the project and task deadlines listed clearly.
    12. receive notifications through email and/or text for approaching deadlines and disable them if I want.
    13. view my level based on the number of tasks I have completed.
    14. easily join a team with either a designated team leader delegating tasks to others or a team without a designated leader democratically deciding what tasks should be done.
    15. view a progress bar visually showing progress made towards the end goal.
    16. (if time allows) view a rough estimation of the time to complete a task.
- As a team member, I would like to be able to:
    1. view tasks to be completed, their point values, and their status.
    2. indicate that I am currently working on a task and then add notes if I feel like I need to share information with my team.
    3. upload files from my computer to my Enjoyagoals group(s) when I have completed a task.
    4. indicate that I have completed a task.
    5. view the number of the tasks that have been completed.
    6. view the tasks that are being completed by other team members.
    7. view the details of each completed task, such as the date & time marked as complete, who completed the task, the point value, etc.
    8. communicate with other team members through a messaging system.
    9. start and participate in a voting system in case the team needs to kick a member.

10. view the leaderboard that shows the other team members' levels & rankings.
11. query the leaderboard.
12. generate a pdf file that includes a progress report for every team member.
13. (if time allows) implement a coup d'etat system to remove the leader.

- As a team member with a team leader, I would like to be able to:
  1. request that the team leader reviews my work & provides either points or feedback.
- As a team member without a team leader, I would like to be able to:
  1. propose tasks to be completed as well as point values for the tasks to be completed.
  2. vote on each task proposal and each point value proposal.
  3. request that the rest of the team review my work.
  4. receive feedback and points for the tasks I have completed.
  5. view a leaderboard showing everyone else's contributions, levels, and rankings.
- As a team leader, I would like to be able to:
  1. invite team members to my team at any point.
  2. create tasks to be completed with the details of each task clearly displayed, such as the date & time marked as completed, who completed the task, the point value, etc.
  3. assign point values to each task created.
  4. provide feedback for every task completed.
  5. view a task's status.
  6. forcibly remove a team member from a task they are currently working on.
  7. easily review any finished tasks to ensure proper completion.
  8. view a leaderboard graphically visualizing each team members' contributions, levels, and rankings.
  9. create announcements that will notify all of the members in my room through email and/or text.
  10. easily transfer my leader role to another member if I leave the room.
  11. assign moderators with less authority than me but more than the other members to my rooms.
  12. edit pending tasks assigned by the rest of the team members before approving or ignoring them.
  13. (if time allows) deduct points from team members if a task was not completed properly.

Non-Functional Requirements:

Architecture and Performance:
The application will have a completely separate backend and frontend. This will allow us to work on the backend and the frontend simultaneously so we can avoid compatibility issues and distribute work efficiently.

The backend will implement a RESTful API (Representational State Transfer) in Node.js and Express. Node.js is a robust framework that offers multiple features to make our web application scalable. Additionally, it is an asynchronous language and is extremely fast compared to most other languages. Express is used to handle various HTTP requests and to write responses to specific URLs. When Express and Node.js are used together, the response time for the application is low.

The frontend will be developed using React and will extract data from the backend by triggering API endpoints. Moreover, development in React is very fast because it uses virtual DOM and reusable components that developers can use. We will be using React's Axios library to send asynchronous HTTP requests to REST API endpoints on the backend.

If the frontend and the backend are independent of each other, the application can always be rewritten in other languages or frameworks and expanded onto other platforms.

Security
Making the application secure and safe is critical as we will store sensitive user information. MongoDB allows us to store data with authentication, access control, and encryption features. This will ensure that sensitive data is stored securely and cannot be exploited easily. We will also ensure that users can only view the project rooms they have access to, not someone else's. Additionally, if a team decides to have a team leader, they will be responsible for adding or removing tasks and determining whether a team member has completed a task. Every request to trigger an API endpoint will be authenticated to ensure user safety.

Usability
The interface should be simple and intuitive while having a fun design reminiscent of a video game. Features should be easy to group into modules, so the interface should not be cluttered with buttons or fields. The leaderboard inside the room should be reasonably straightforward to understand and interact with. Looking at certain rooms' leaderboards while not in any room is done by simply hovering over the room on the room selection interface. Our application will be usable on all devices with no significant change in design.

Hosting/Deployment
The backend and frontend will be developed independently at the same time, and they will also be deployed independently. The node server on the backend will be deployed on Heroku, and the frontend React will be deployed on Netlify.