📌 Sprint Backlog

Sprint 1:

Before the Sprint the team came together to discuss the overall Tech Stack and strengths and weaknesses of each team member. The most imperative step prior to starting on the project was the design of the product.
Meeting with the client and understanding their needs and the audience of the system allowed us to come up with user stories and a product requirement document 📑 User Stories & Product Requirements . For the first Sprint, the workload was allocated between the group members. Our Front-end Team consisted of Nanda, Thomas and AJ while the back-end Team consisted of Riley and Jonas.

Frontend:

- UI interface design (whole team)
  - Home : Riley
  - Profile: Thomas
  - Saved Problem: Nanda
  - Categories: AJ
  - Workspace: Jonas
- Home Page Interface (Nanda)
- Profile Page Interface (Thomas)
- Problem Categories Page Interface (AJ)
- Problem Page (Nanda)
- drag and drop section (Nanda)
- Categories Section (AJ)
- Past problems Page (Nanda)

Backend:

- User IP based Statistics
- Admin API Routes (Authentication)
- Routes for AI API Prompt Interface generating Parsons Problems
- Routes for feedback from IDE
- Problem Prompt(description)

Sprint 2:

In this sprint, our team is focusing on finalizing the backend and frontend for the team to move on to the next stage of connecting frontend with backend.

- past problems page
- Context for each category back and front end
- applying cookies for data management
- Routes protection
- Routes for changing account details
- User Login Logout
- Parsons Problems Base Statistics
- Add on to agile ceremonies

Sprint 3:

In this sprint, our team will focus on finalizing the integration between the front-end React app and the back-end Express API, ensuring seamless communication and functionality across the two systems. The primary goal is to implement key API endpoints in the Express app that will power dynamic content and data fetching in the React interface. Additionally, we will connect the front-end components to these APIs, enabling a fully functional, user-friendly interface. Comprehensive testing, both unit and integration, will be conducted to ensure a smooth and efficient user experience upon release.

- Admin API Routes
- Connecting frontend with backend
- Improve validation of Gemini responses
- Contexts for each category for back & front end
- Apply indentation mechanism to drag and drop
- Utilizing a working compiler in the backend to run python code