Team 39 Sprint 1 Retrospective

Enjoyagoals

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

What went well?

In general, we completed the majority of the backend algorithms and the UI for over half of the user stories. Users can register for an account, log in, create tasks, view their profile, edit their profile picture, and also edit their background image.

User Story #1:

As an individual user, I would like to be able to register for an Enjoyagoals account.

#	Description	Estimated Time	Owner
1	Create UI panel for user registration	2 Hrs	Ryan, Prahas
2	Create an algorithm to obtain all the necessary information from the new user to successfully register	2 Hrs	Karim, Nabi, Nick
3	Debug and test the algorithm that obtains all the necessary information from the UI	3 Hrs	Karim, Nabi, Nick
4	Create an algorithm to send the new user data to the database	2 Hrs	Karim, Nabi, Nick
5	Debug and test the algorithm that sends the data to the database	3 Hrs	Karim, Nabi, Nick
6	Connect the algorithms to the UI so new users can register successfully	2 Hrs	Karim, Nabi, Nick

Completed:

New users can use the registration page to begin the registration process. During registration, we are checking if the email address and the username that the new user enters is taken or not. After that if the new user successfully makes an account, they receive an email confirmation to the email address they used during registration.

User Story #2:

As an individual user, I would like to be able to login and manage my Enjoyagoals account, and reset my password.

1	Create UI panel for login screen & account management screen	3 Hrs	Ryan, Prahas
2	Create UI screen for resetting password	2 Hr	Ryan, Prahas
3	Test and debug calls to backend	2 Hrs	Karim, Nabi, Nick
4	Create algorithm to manage requests for login and account management	4 Hrs	Karim, Nick, Nabi
5	Implement hashing function to handle passwords in database & login attempts	3 Hrs	Nick
6	Create algorithm to send a password recovery email to user	3 Hrs	Karim, Nabi, Nick
7	Test and debug backend algorithms	5 Hrs	Karim, Nabi, Nick

Completed:

There is a UI for logging in and registering that sends inputted data to the backend through HTTP POST requests. The backend has routes to handle each scenario and does format validation & ensures all required fields are filled in. The backend route for registering hashes the user's password before saving all the data into a new User object in the database. For logging in, there is an option for people who have forgotten their password, and there is a backend algorithm that sends an email to allow users to reset their password.

User Story #4:

As an individual user, I would like to be able to easily set a custom profile picture and customize my profile page's theme and banner.

1	Find set of default profile pictures	0.5 Hrs	Ryan, Prahas
2	Implement ability to upload any image as a profile picture	3 Hrs	Ryan, Prahas
3	Find set of colors for background to use	0.5 Hrs	Ryan, Prahas
5	Implement new profile picture and background	1 Hrs	Ryan, Prahas

Completed:

When the user clicks a button next to their profile picture, they can upload a profile picture of their choice from the files on their computer and see it change in real time. They can change the background of their displayed profile page by selecting one of 3 colors from a dropdown menu. All of these changes are retained even when the user refreshes the page or closes then opens it.

User Story #5

As an individual user, I would like to be able to display my contribution to the room I am currently in, my email account and phone number on my Enjoyagoals account alongside other personal information for others to see.

1	Create UI to display all the personal information	4 Hrs	Ryan, Prado
2	Create an algorithm to obtain all the information you need to display	3 Hrs	Karim, Nabi, Nick
3	Debug and test algorithm	3 Hrs	Karim

Completed:

There is a UI that displays personal information and the backend has routes to handle queries to retrieve the personal information for every user. Each profile displays the user's email account, phone number, and contributions.

User Story #6

As a team leader, I would like to be able to invite team members to my team at any point.

1	Create a user controller to see if a team member is a leader	0.5 Hrs	Nabi, Nick, Karim
2	Create algorithm to generate a unique link for invitation	4 Hrs	Nabi, Nick, Karim
3	Expire the link after the specific period of time or if the user joined the team	3 Hrs	Nabi, Nick, Karim
5	Debug and test algorithm	4 Hrs	Nabi, Nick, Karim

Completed:

If a team leader is trying to invite another user, they can create an invitation link. The default link will expire in 1 day and only 1 person can use it. After 1 day or after using it by a user once, the link will expire and it will no longer be used. Users can change these 2 categories by editing the link.

User Story #7

As a team leader, I would like to be able to create tasks to be completed with the details of each task clearly displayed, such as the date & time created, who is assigned the task, the deadline, the point value, etc.

1	Create controller logic for task creation	3 Hrs	Nabi, Nick, Karim
4	Create a user controller to see if a member of team is a leader or not	1 Hrs	Nabi, Nick, Karim
5	Add the task details with a unique id to the database	2 Hrs	Nabi, Nick, Karim

Completed:

When the team leader clicks the create task button, they are able to fill in the details of a task with a task name, task description, task difficulty, assigned user, specific deadline and a point value. If any space is left blank on a task creation page, the controller logic won't allow the leader to add the details to the database. Each task has a unique id so that it would be possible to differentiate the team member to whom the task was assigned.

User Story #8

As a team member without a team leader, I would like to be able to propose tasks to be completed as well as point values for the tasks to be completed.

2	Create an algorithm to send the proposed task to the database	2 Hrs	Nabi, Nick, Karim
3	Debug and test algorithm	4 Hrs	Nabi, Nick, Karim

Completed:

Backend has the necessary schema and routing to accept and store a task and all its related data.

User Story #9

As an individual user, I would like to be able to view all of the tasks I need to finish in a comprehensive list after logging in.

2	Create algorithm to obtain information about each task from the database	3 Hrs	Nabi, Nick, Karim
3	Debug and test algorithm with calls to the backend	4 Hrs	Nabi, Nick, Karim

Completed:

Backend has the necessary schema and routing to store task data in the database and then retrieve & send it upon request from the frontend.

User Story #10

As an individual user, I would like to be able to easily check the details of each individual task (description, date created, deadline, point value).

2	Create algorithm to obtain information about the task from the database	4 Hrs	Nabi, Nick, Karim
3	Debug and test algorithm	4 Hrs	Nabi, Nick, Karim

Completed:

Backend has the necessary schema and routing to store task data in the database and then retrieve & send it upon request from the frontend.

User Story #11

As a team member, I would like to be able to view the tasks to be completed, their point values, and their status.

2	Create an algorithm to obtain information about all the tasks	4 Hrs	Nabi, Nick, Karim
3	Debug and test the algorithm	4 Hrs	Nabi, Nick, Karim

Completed:

Backend has the necessary schema and routing to store task data in the database and then retrieve & send it upon request from the frontend.

User Story #12

As an individual user, I would like to be able to join, switch between, or withdraw from different project rooms at any time after signing in.

5	Create an algorithm so the database is updated whenever a user joins a new room or withdraws from a room	2 Hrs	Nabi, Nick, Karim
6	Test and debug algorithm	3 Hrs	Nabi, Nick, Karim

Completed:

Backend has the necessary schema and routing to store which Rooms a User is in in the database, store the data for each room in the database and then retrieve & send all of this upon request from the frontend.

User Story #13

As a team member, I would like to be able to 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.

4	Create an algorithm to send the new task list to the database	2 Hrs	Nabi, Nick, Karim
5	Debug and test the algorithm to send the new list to the database	3 Hrs	Nabi, Nick, Karim

Completed:

Backend has the necessary schema and routing to store task data in the database and then retrieve & send it upon request from the frontend. Additionally, Room objects have an array of Tasks[] and this array can be queried and sent to the frontend upon request as well.

What did not go well?

In general, most of the frontend functions were not completed while the backend functions were, so many of our user stories could not be finished even when they were functional on the backend.

User Story #3

As an individual user, I would like to be able to register using my google account if I do not have an Enjoyagoals account.

1	Add the “register using google account” option to login screen	0.5 Hrs	Ryan, Prahas
2	Create algorithm to interface with Google Login API	2 Hrs	Karim, Nabi, Nick
3	Create algorithm to populate database with user information from linked Google account	2 Hrs	Karim, Nabi, Nick
4	Test and debug algorithms	3 Hrs	Karim, Nabi, Nick

Not Completed:

We decided to not implement this user story during this sprint because it took too long to implement due to lack of prior experience on backend development and we were short on time.

User Story #4

As an individual user, I would like to be able to easily set a custom profile picture and customize my profile page's theme and banner.

4	Implement buttons to change profile picture and background color	2 Hrs	Ryan, Prahas

Not Completed:

Though we have found a set of default profile pictures for the user to select, we have yet to implement the functionality to actually use them on the profile page. As of now, they are displayed in a static popup menu and cannot be clicked.

User Story #6

4	Create frontend view for invitation link	1 Hrs	Ryan, Prahas

Not Completed:

The current system sends the invitation link directly to the user's email. No frontend has been developed for this yet.

User Story #7

2	Create frontend view for task post	3 Hrs	Ryan, Prahas
3	Make the created task visible on main feed	2 Hrs	Nabi, Nick, Karim

Not Completed:

Tasks postings do not appear on the main feed, only on the server. The frontend view still needs to be developed.

User Story #8

1	Create UI to create a new task and propose it	3 Hrs	Ryan, Prahas
4	Connect the algorithm to the UI to display all the information	2 Hrs	Nabi, Nick, Karim

Not Completed:

There is an existing UI for creating new tasks, but it is extremely rough and still needs to be polished. New tasks that were created cannot be cannot be displayed on the UI yet.

User Story #9

1	Create UI panel to display every task in a list	2 Hrs	Ryan, Prahas
4	Connect algorithm to UI that displays each task	2 Hrs	Nabi, Nick, Karim
5	Implement button to show only tasks user needs to finish	1 Hrs	Nabi, Nick, Karim

Not Completed:

The UI to display each task in a list and the button to filter all but the tasks the user needs to finish have not been developed yet, so the algorithm cannot connect to it.

User Story #10

1	Create UI to display detailed information about a task	1 Hr	Ryan, Prahas
4	Connect algorithm to UI that displays information about the individual task	1 Hrs	Nabi, Nick, Karim

Not Completed:

The frontend for displaying detailed information about tasks has not been created yet, so the algorithm cannot be connected to it to display information about the task.

User Story #11

1	Create UI to view all the tasks that need to be completed, their point values, and their status	1 Hrs	Ryan, Prahas
4	Connect UI to algorithm that displays all the tasks to be completed and all the relevant information	4 Hrs	Nabi, Nick, Karim

Not Completed:

The frontend for displaying all the user tasks still needs to be created and connected to the existing algorithm to display tasks to be completed.

User Story #12

1	Create UI to make a user input a unique room code or accept an invitation to join a room	2 Hrs	Ryan, Prahas
2	Create UI so users can switch between rooms	2 Hrs	Ryan, Prahas
3	Create UI so users can withdraw from rooms	2 Hrs	Ryan, Prahas
4	Create an algorithm so users can join, switch between, and withdraw from rooms	3 Hrs	Nabi, Nick, Karim
7	Connect the algorithm to UI so users can join, switch between, and withdraw from rooms	2 Hrs	Nabi, Nick, Karim

Not Completed:

The frontend for displaying joining, switching rooms, and withdrawing from rooms has not been created yet, so the algorithm cannot be connected to it to display information about the user joining, switching rooms and withdrawing from rooms.

User Story #13

1	Create UI to indicate that a task is currently being worked on by a team member	2 Hrs	Ryan, Prahas
2	Create an algorithm to update the task list so the task a team member chooses cannot be completed by someone else	2 Hrs	Nabi, Nick, Karim
3	Debug and test the algorithm to update task list	3 Hrs	Nabi, Nick, Karim
6	Connect all the algorithms to the UI	2 Hrs	Nabi, Nick, Karim

Not Completed:

The frontend to indicate whether a task is being worked on by a team member has not been created. The backend code to update the list to stop multiple members from attempting the same task has also not been written. The backend to update the list of tasks has also not been debugged or connected to the UI.

How should you improve?

The first thing that we can improve on for the next sprint is distributing acceptance criteria better so that not every point is dependent on both frontend and backend. We should split the points that would depend on both frontend and backend into separate parts, as this will allow us to be more precise with what it is we're trying to accomplish. This will probably result in us having more acceptance criteria.

The second thing we can improve on for the next sprint is organizing what needs to get done & in what order. During this past sprint, it seemed like the team knew generally what we wanted to accomplish, but there was a bit of uncertainty of what specifically needed to be done since we discussed making progress in terms of completing User Stories. We failed to consider the fact that some of our User Stories had some overlap since they required the same foundational components. In week 3, we tried listing out exactly what each of our 2 teams needed to do (ex: which schema and routes the backend needed to implement, which UIs and requests the frontend needed to construct). This seemed to be helpful so we will try doing this at the beginning of the next sprint, and if it continues to be useful we will do this on a weekly basis.