

Stellar Sync

The Astrologists (the Team)

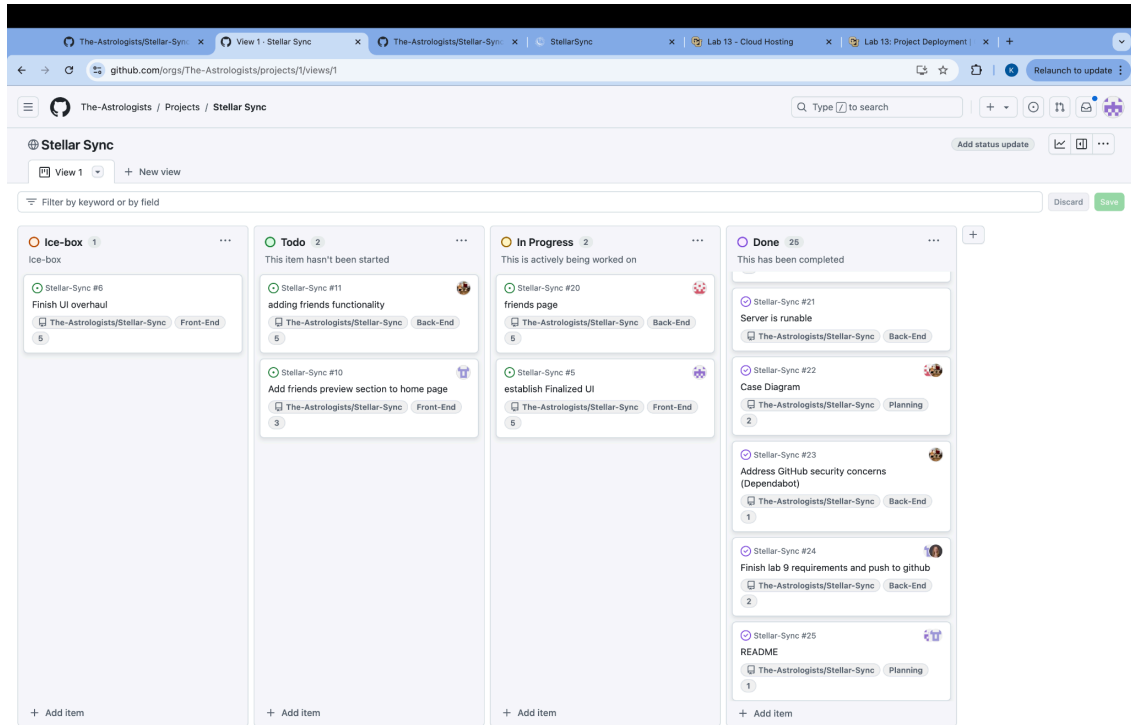
- Katie Miller: k-miller8
- Brooke Seifert: brookeseifert
- Brooke Nelson: bnelsonCU27
- Sam Kutcipal: samkuts
- Aleqz Baker: azb333

Project Description

Stellar Sync is a celestial themed application which provides users with insights about their zodiac sign and other personalized topics based on their birthdate. Individuals seeking personalized recommendations and affirmations rooted in astrology will be provided with ample information about themselves and their friends. Upon registration, users provide their birthdate which is then used to generate information about their zodiac sign. From this, users can view their account information including a brief description of the attributes of their sign, a personalized affirmation, and three song recommendations specifically curated for the user. The daily affirmation and songs are both generated based on the user's specific signs through calls to the OpenAi API on every login. Beyond profile information, users can navigate to a search page to search for other users based on username and have the option to become friends with them. Added friends will then be displayed in a list with basic profile information on the user that added them's friends page titled friends. As a whole, the application provides users with personalized zodiac information across a seamless interface with an artistic cosmic presentation.

Project Tracker: GitHub Project Board

- Link: <https://github.com/orgs/The-Astrologists/projects/1/views/1>



Video

- <https://youtu.be/Thr76zTjSkw>

VCS

- Link to your git Repository: <https://github.com/The-Astrologists/Stellar-Sync>

Contributions

Katie: Updated the README page with contributors and default requirements. Created a working CSS style page across files and set default styling and color scheme. Added multiple default users with complete profiles to the database for use and wrote code to print user profile information on the home page. Added functionality to prevent logged-in users from logging in and non-registered users from logging out. Accounted for attempting to create duplicate usernames when registering.

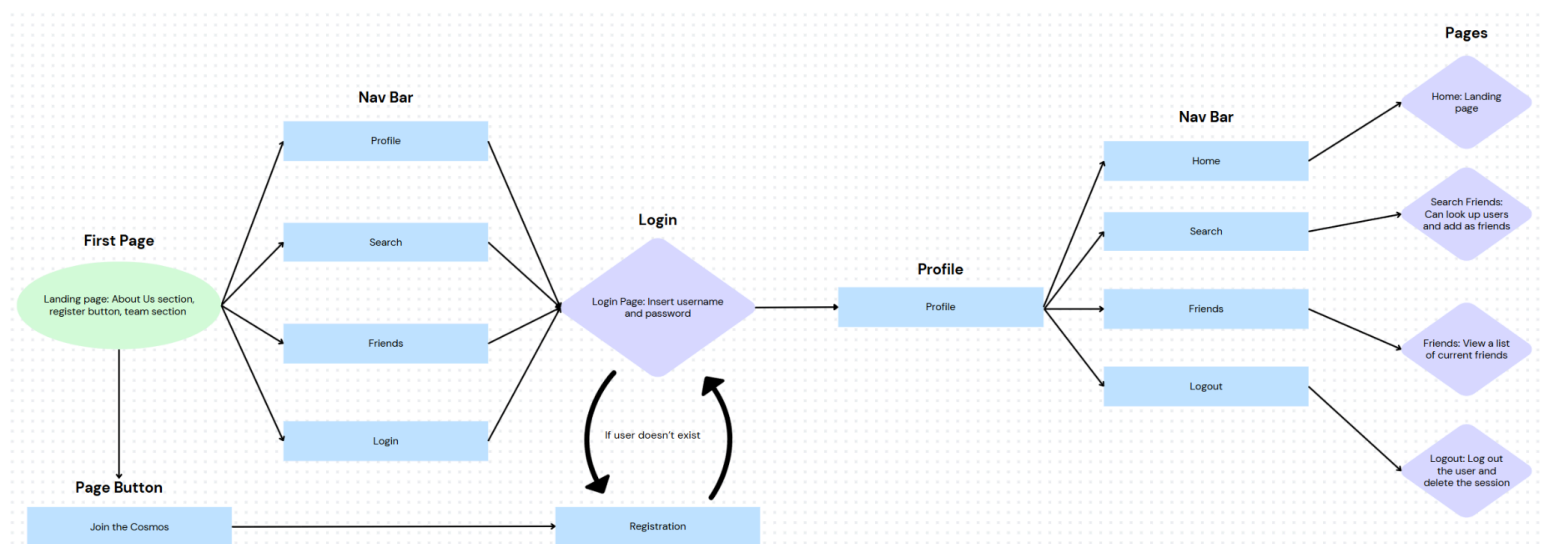
Brooke S: Created and styled the profile page, including divs for the profile picture, horoscope, daily affirmation, song recommendations, and access to friends page. Added columns to the users table and appropriately managed the insert.sql file for database testing. Created getSign function to assign a user's zodiac upon registration. Styled login and registration pages and added user-friendly messages for incorrect passwords, etc. Created finalized wireframes for all pages. Implemented the API for daily affirmation that refreshes based on login.

Brooke N: Defined the original file/directory structure required by the project. Created and styled the landing page, navigation bar, footer, and the search page. Coded the relevant queries and functions on the search page that allows users to search other users and add them as friends (doesn't allow users to look up their own user and informs the user if they are already friends with a user). Fixed a few styling bugs and database issues. Added a tab icon specific to our project theme.

Sam: Created the initial registration page with appropriate fields for username, name, password, and birthday to log user information to the database. Contributed to the function of and wrote the first version of the friends page. Created the initial song request page, calling the API and displaying songs. Worked on general bug fixes and debugging the friends page issues. Went through code and commented all functions and pages to ensure clarity and usability of all code. Spearheaded communication by making a discord server for team members to communicate with each other and share code and ideas.

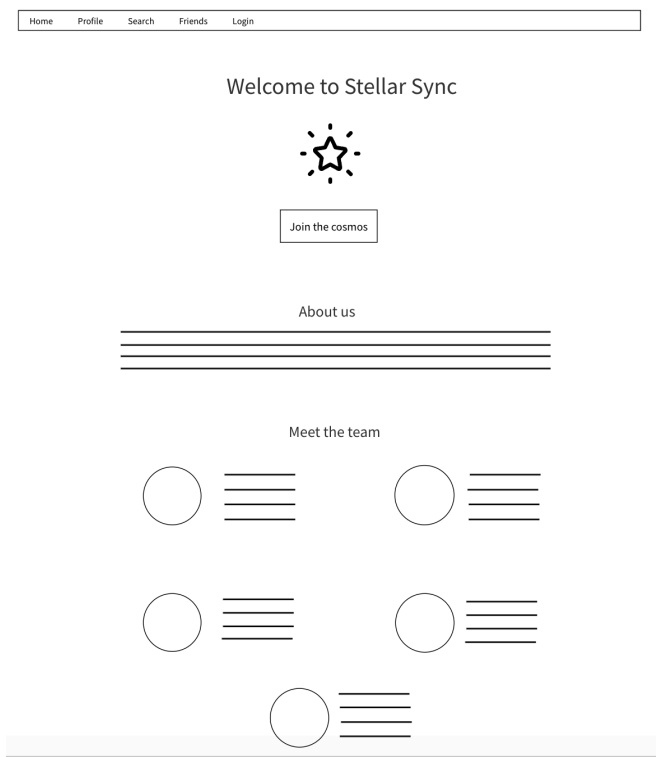
Aleqz: Set up the initial repository and local deployment. Initiated and incorporated the OpenAI API and guided its implementation. Created the friendships table and ensured users could not add themselves as friends. Created and implemented full functionality of the horoscope description array based on user sign. Implemented the song recommendation API functionality for three individual songs. Rigorously set up and debugged the website server for final deployment on Render servers.

Case Diagram




Wireframes

- Landing Page



● Profile Page

[Home](#) [Profile](#) [Search](#) [Friends](#) [Login](#)

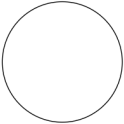


Name:

Birthday:


Sign:


About you




Your daily horoscope


Some music for you today...







Say hello to your friends across the galaxy



● Login Page

[Home](#) [Profile](#) [Search](#) [Friends](#) [Login](#) [Logout](#)

LOGIN

username

password

Login

register

- Friends Page

Home	Profile	Search	Friends	Login
------	---------	--------	---------	-------

Create Your Constellation of Friends

Friend	⌵
Friend	⌵
Friend	⌵
Friend	⌵
Friend	⌵
Friend	⌵

- Search Page

Home	Profile	Search	Friends	Login
------	---------	--------	---------	-------

Create Your Constellation of Friends

Search	⌵	Search
		Add Friend
		Add Friend

- Register Page

Home	Profile	Search	Friends	Login	Logout
------	---------	--------	---------	-------	--------

REGISTER
Username
First name
Last name
Password
register

Test Results

Test Case 1: Valid User Login

Objective: Users can successfully login with valid information

Preconditions: User has already successfully registered an account

Steps:

- Go to the Login page
- Input registered/valid username and password
- Click Login button

Expected Result:

- User should be redirected to their profile page with all of their information

Acceptance Criteria:

- All steps are completed without errors

Test Environment:

- Localhost / console

Test Outcome:

- Pass or fail

Test Case 2: Render Login

Objective: Login page is successfully rendered upon redirecting to login

Precondition: Login page is successfully rendered open opening the site

Steps:

- Open site
- Navigate to login page by clicking on login

Expected Result:

- User should be redirected to the login page from the landing page upon clicking on login

Acceptance Criteria:

- The page is rendered successfully
- All fields and buttons visible and working

Test Environment:

- Localhost / console

Test Outcome:

- Pass or fail

Test Case 3: open AI api call rendering

1. **Objective:** Verify that users can successfully generate a daily horoscope
2. **Preconditions:** User is registered and logged in.
3. **Steps:**

- Go to the profile page.
 - Find the horoscope section of the page
 - Be able to click the generate button
 - OpenAI api call that generates horoscope
4. **Expected Result:**
- User should be able to navigate to profile page.
 - Be able to click the generation button
 - User should be able to view horoscope
5. **Acceptance Criteria:**
- All steps are completed without errors.
 - Horoscope can be viewed
6. **Test Outcome:**
- Horoscope can be seen
 - LocalHost3000

Test Case 4: Render Landing

Objective: Landing page is successfully rendered upon opening the site

Precondition: Loading the site

Steps:

- Open site

Expected Result:

- User should be redirected to the landing page upon opening the site

Acceptance Criteria:

- The page is rendered successfully
- All fields and buttons visible and working

Test Environment:

- Localhost / console

Test Outcome:

- Pass or fail

Test Case 5: Testing Landing

Objective: Users can successfully access landing with valid information

Preconditions: Loading the site

Steps:

- Go to the Landing page

Expected Result:

- User should be able to access the landing page

Acceptance Criteria:

- All steps are completed without errors

Test Environment:

- Localhost / console

Test Outcome:

- Pass or fail

User Observations

- When testing our application with users, we saw the users explore the landing page first, then register an account using the button on the landing page. After being directed to the login page, they logged in successfully and then explored the rest of the pages. The users' reasoning for their actions was that they knew they had to register an account first before being able to log in (they didn't click on other pages, they directly went to "Join the Cosmos"). Their behavior is consistent with the use case because we would assume that a new user on a website would intuitively register a new account before attempting anything else. We did not observe any deviation from the expected actions, so we did not make any corresponding changes.

Deployment

- <https://stellar-sync-hggg.onrender.com/landing>