

BetWise

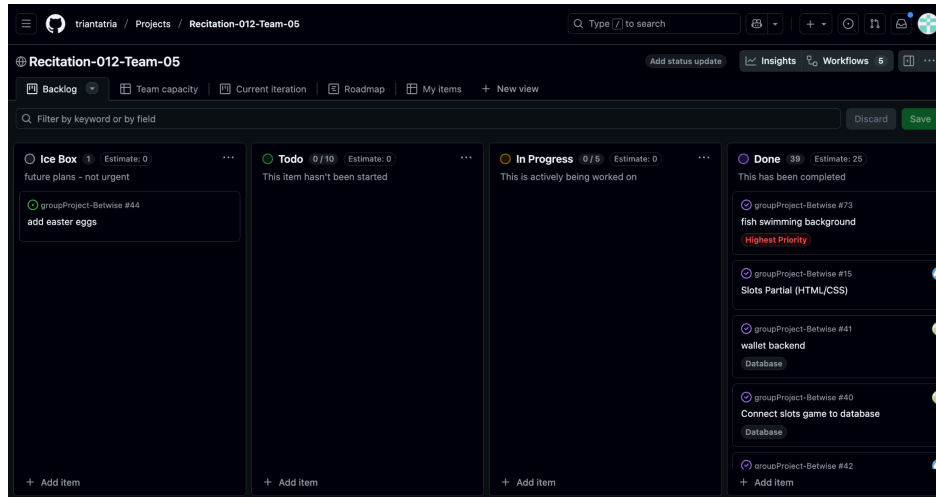
Created by Jackie Auerbach (jackieauerb), Connor Edelheit (triantatria), Nathan Megersa (nmegersa), Adam Wagner (AdamWagner731), and Linley Denslow (lide4293).

Project Description:

BetWise is a web-based casino designed to make gambling safe, fun, and educational, without the risk of losing all of your hard-earned money. It features a sleek interface, smooth animations, and three exciting games. BetWise gives users the thrill of casino gaming while protecting their wallets and their pride. Its features include three polished games to enjoy: Blackjack, Mines, and Slots. These games are created for quick sessions, responsive design, and engaging gameplay. Our application includes a credit-based system, with in-game credits, ensuring an entertaining and risk-free experience. The website is built to look and feel great with instant feedback. BetWise's target audience includes beginner gamblers wanting to learn how casino games work, users looking for a fun, stress-free way to play, and individuals who want the thrill of gambling without financial loss or debt. Another feature that is included is a leaderboard which displays other players ranked by their wins in the games. This adds an exciting competitive edge to BetWise. We also have a wallet page, where you can deposit credits and keep track of your earnings and bets. For the future of BetWise, we would like to include separate leader boards and an AI tutoring/ tips feature.

Github Project Board Link and Screenshot:

<https://github.com/users/triantatria/projects/2/views/1>



Video Demo:  Screen Recording 2025-12-05 at 3.06.33 PM.mov

Github Repository Link:

<https://github.com/triantatria/groupProject-Betwise>

Contributions:

Linley:

My main tasks throughout this project were focused on the slots page and the profile page. For the profile page, I set up the path, the UI, added the logout button, and made sure it successfully displayed user information. For the slots page, I was focused on the logic of the game and the animation. I also made sure that the credits were properly updated, and figured out the calculations needed for your earnings. I also created many story points for the project board and made sure they were delegated to teammates for our weekly executables, and created the ReadMe.

Jackie:

I built the CSS styling, animations, transitions, and overall user experience of BetWise. Early on, I created and styled the Handlebars pages, partials, and CSS framework for the login/registration flow, homepage, Mines, Blackjack, Slots, the wallet, and the leaderboard. This gave us a shared visual starting point that made testing and refining game logic significantly easier. I set up reusable animations and consistent

styling across the pages. As backend features were developed, I integrated the UI with the game logic so that the DOM updated smoothly. Overall, I was responsible for the user experience layer and tying it to the backend.

Nathan:

I took notes for weekly TA meetings in Google Docs and helped draft weekly release notes using generative AI (GPT-5). I helped design and implement the PostgreSQL database for user information and transactions, and connected the games and wallet to the backend so deposits, bets, and wins update correctly. I also had some trouble at first with backend queries and front-end synchronization but worked through it. Finally, I helped write Chai tests to validate user credentials during registration and login (Lab 10).

Connor:

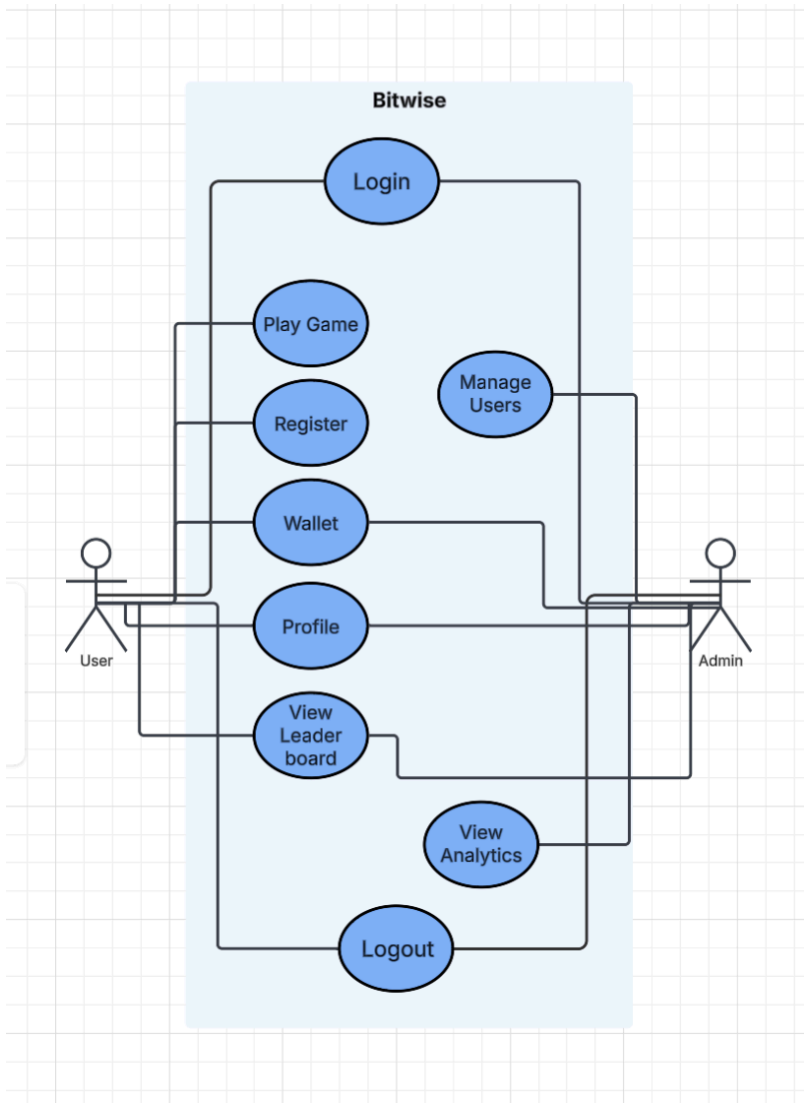
I built the initial API Paths and Login/Register page which were then iterated upon. I connected the profile page and the leaderboard page to the backend using SQL queries and passing values through render statements. I also worked on the form for the register page and the insert SQL query used to add new users to the database.. Finally, I created the PostgreSQL database on Render, created the web service on Render, configured the web service settings, set up the environment variables, set up the .env file, deployed the application, and initialized the database (Lab 12).

Adam:

I was responsible for various tasks in logic and styling. My first main contribution was getting blackjack and mines logic coded and creating partials for these two games. My initial code didn't work perfectly right away so I spent a few weeks updating and making sure the logic to the games was correct. This included blackjack card dealing functions and randomization of the mines on the board. I also added

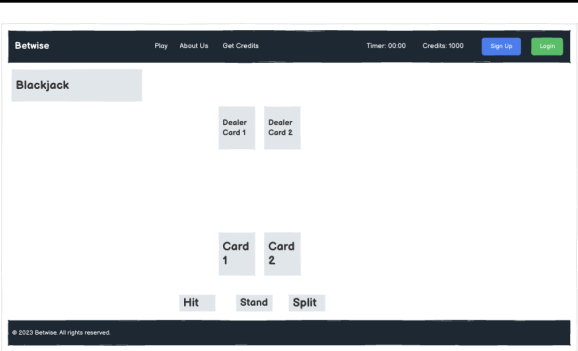
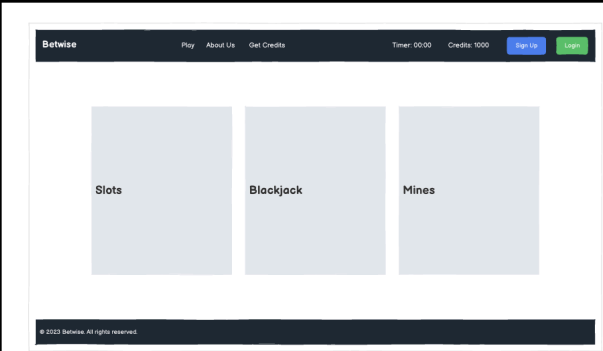
RegEx requirements for passwords to the register page. Finally I added images for each of our games using generative AI to generate images of fish playing each game.

Use Case Diagram:

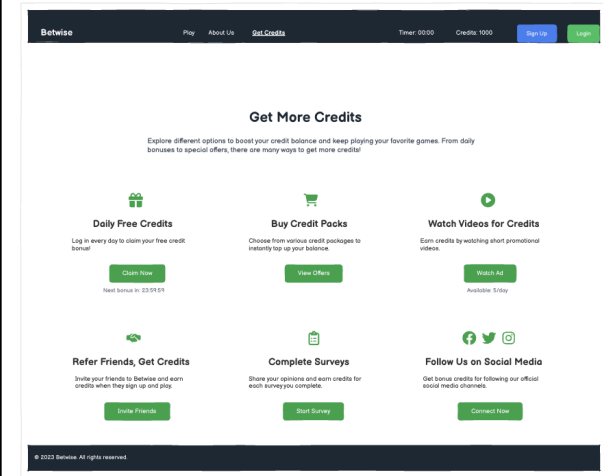


Wireframes:

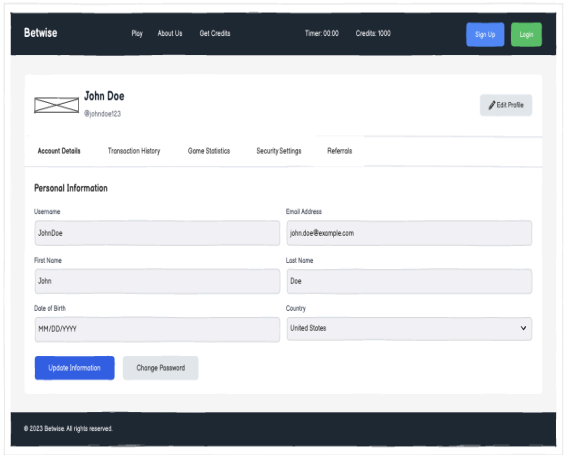
Home Page:	Blackjack:
------------	------------



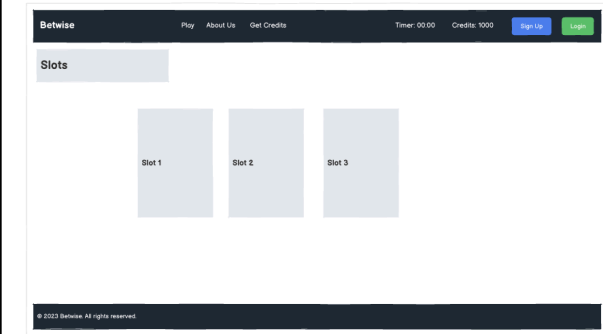
Wallet:



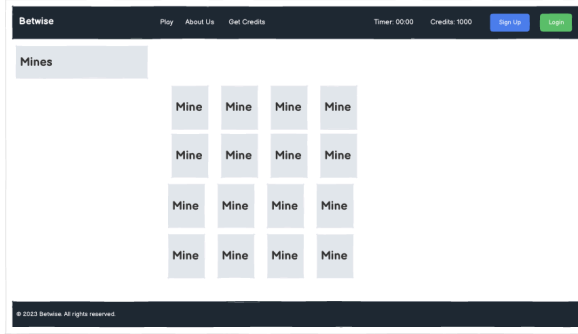
Profile:



Slots:



Mines:



Login Page:

Leaderboard:

<p>Register Page:</p>	<p>About page:</p>

Test Results:

Test user: Hialie H.

Test Case 1: Successfully registering an account and logging into the application

- What are the users doing?
 - Users used the registration page to create an account by entering first/last name, email, username, and password. They also tried invalid emails and passwords that didn't meet the regex requirements, and then tested the login page with both unregistered and registered credentials.

- What is the user's reasoning for their actions?
 - They wanted to verify that registration only succeeds with valid credentials (valid email and regex passing password) and that the login page only allows access when the username and password match an existing user in the database.
- Is their behavior consistent with the use case?
 - Yes, their behavior aligns with the intended use case of the registration and login pages: creating accounts with valid data, storing them in the database, and then logging in with those credentials. Trying both valid and invalid inputs reflects how real users would interact with these features and helps confirm validation and authentication.
- If there is a deviation from the expected actions, what is the reason for that?
 - Earlier, users asked about being able to register their name and email, in addition to username and password, before those fields existed on the registration page. They wanted to do this to add more user information and possibility of emails/newsletters being sent to users in the future.
- Did you use that to make changes to your application? If so, what changes did you make?
 - We did use that to make changes to our application, deciding to add first/last name and email fields that users could fill out when registering that would be stored in the database, along with their username and password once they signed in.

Test Case 2: Adding Credits to User Balance from Wallet Page

- What are the users doing?
 - Logged-in users navigate the site to find where to add credits, then go to the wallet page. They test different deposit amounts to see how many credits they can deposit at once and what happens with invalid inputs (non-numbers, negatives, or amounts outside 1–1000).
- What is the user's reasoning for their actions?

- They are checking that the wallet correctly updates the user's balance based on the deposit amount and that invalid inputs are rejected (non-numeric, 0, negative, or out of range).
- Is their behavior consistent with the use case?
 - Yes. The wallet's purpose is to display the correct balance and update it when users deposit credits, while enforcing limits and input validation. Their tests mirror how users would manage balances and deposits in a real application.
- If there is a deviation from the expected actions, what is the reason for that?
 - Users noticed there was no daily limit on deposits. They realized a user could keep depositing the maximum (1,000 credits) repeatedly and effectively give themselves unlimited credits, which could send the wrong message about gambling having no financial limits.
- Did you use that to make changes to your application? If so, what changes did you make?
 - Yes. We added a daily credit limit of 5,000 credits and a `last_credit_topup_date` column in the users table. After reaching the limit, users cannot deposit more credits until the next day. This prevents "infinite" credits and reinforces that money/credits are limited.

Test Case 3: Mines game functionality, placing bet, playing game, round result

- What are the users doing?
 - Logged-in users choose to play the Mines game from the home screen. On the Mines page, they see a 5×5 grid, their balance, bet input, mines input, and "Start Game" / "Cash Out" buttons. They try different bet amounts to confirm they can only bet up to their balance, verify that clicked tiles are correctly marked safe, and then observe how their balance changes on both the Mines page and wallet after winning, losing, or cashing out.
- What is the user's reasoning for their actions?

- They are checking that they cannot bet more credits than they have, that the game logic works correctly for safe tiles, mines, and cashouts, and that the balance updates correctly after each round outcome.
- Is their behavior consistent with the use case?
 - Yes, it is consistent. The Mines game is designed for users to place bets, reveal tiles, and either win, lose, or cash out according to the game rules. Testing bet limits, tile behavior, and balance updates matches expected real-world usage.
- If there is a deviation from the expected actions, what is the reason for that?
 - Users wanted cashout to return more than just the original bet, with the payout depending on how many safe tiles they had revealed. This would introduce more risk and strategy: continue for a higher cashout or stop early.
- Did you use that to make changes to your application? If so, what changes did you make?
 - Yes. We updated the cashout logic so the cashout amount increases after each safe tile based on the user's bet, number of mines, and number of safe tiles revealed. This created a more dynamic game where users can see a changing cashout value as they play, instead of only getting their bet back.

Deployment Link: <https://groupproject-betwise-zh96.onrender.com/>