

WorNDly - Manor-Quad Final Project

Team Member Full Name	NetID
Emily Colleran	ecoller2
Michael Hines	mhines2
TJ Weber	tweber4
Zach Sellman	zsellman

Persistent Storage Design

Our project's storage design is centered around a set of two main tables, as shown in Figure 1: one for storing user credentials like usernames, emails, and passwords, and another for recording gameplay history, including details like total games played, number of guesses and wins, and dates. These tables form the backbone of our system, with Django's authentication system providing added security for user logins. Additionally, we have integrated a REST API to keep track of user coin balances, ensuring a seamless experience across the board. Together, these elements create a sturdy foundation for managing user data and enhancing the overall user experience.

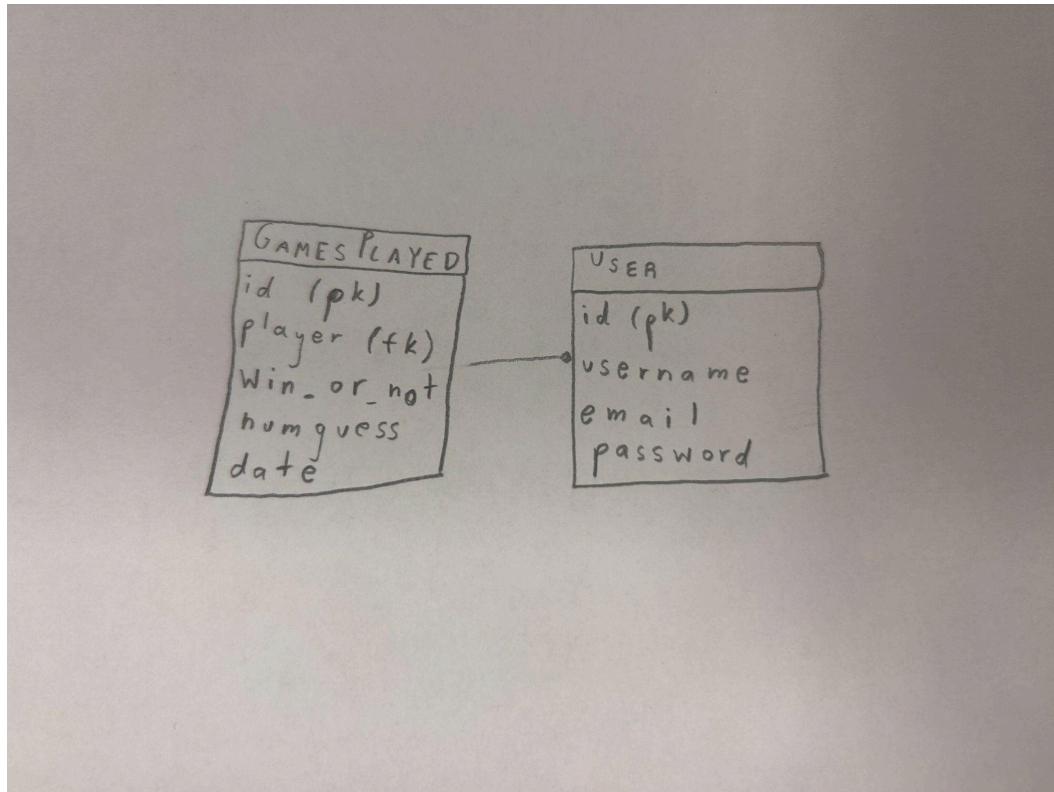


Figure 1 database schema

Demonstration of the Features

Introducing “worNDly,” a dynamic gaming web application focused on delivering an immersive experience through four core features. Users can seamlessly authenticate with sign-up, sign-in, and log-out functionalities (Feature 1). Once logged in, they embark on a thrilling word-guessing challenge, aiming to decipher a 5-letter word within six attempts, aided by color-coded clues (Feature 2). To sustain excitement, the platform limits free daily plays to three while offering additional plays for purchase using Krato\$Coin (Feature 4). The player’s dashboard provides insights into past plays, statistics, and purchase history, enhancing user engagement and strategy (Feature 3). Powered by Python, Django, HTML, CSS, and JavaScript, worNDly promises a seamless and captivating gaming experience. Below in Figure 2 lies our homepage.

In Figure 2, you will find our homepage, featuring a captivating image of Shrek and extending a warm welcome to users. Here, users are presented with the choice to either create a new profile or sign in. Additionally, an exciting easter egg awaits users who click on the “worNDly” text at the top of the page.

Homepage:

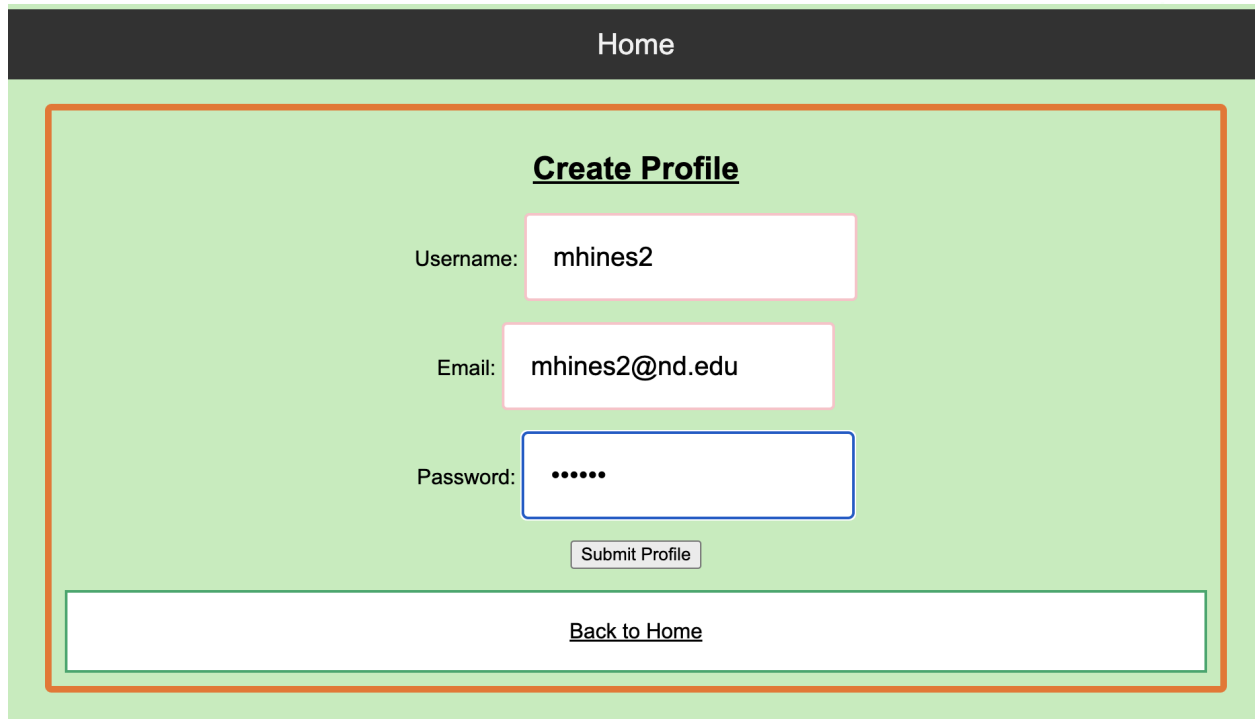


Figure 2 screenshot for homepage showing create/sign in profile

Feature 1.1

In Figure 3, Feature 1.1 enables users to generate their player profiles by entering essential details, such as their username, email, and password. Upon completing the profile creation process and clicking the “Submit Profile” button, the newly entered information is

transmitted to our project database. Subsequently, users can easily navigate back to the homepage by clicking the “Back to Home” button.



The screenshot shows a web interface with a dark grey header bar labeled "Home". Below the header is a light green rectangular area containing a form. The form is titled "Create Profile" in bold black text. It has three input fields: "Username:" with the value "mhines2", "Email:" with the value "mhines2@nd.edu", and "Password:" with masked characters ".....". Below these fields is a "Submit Profile" button. At the bottom of the form area is a white rectangular button labeled "Back to Home".

Figure 3 screenshot for feature 1.1 showing create new user profile

In the event that a chosen username is already in use, as depicted in Figure 4, the profile creation page resets, preventing the transmission of the new user’s data to the database. Instead, users are promptly notified with an error message stating “Username is already taken.”

Back to Home

Username is already taken.

Figure 4 screenshot for feature 1.1 showing unique username system

Feature 1.2

Feature 1.2 enables users to log in to their profiles by providing their username and password. Figure 5 displays our account login interface, where users can enter their credentials.

WORLDLY

a manor quad game

Home

Sign In to Account

Username: mhines2

Password:

Sign in

[Back to Home](#)

Figure 5 screenshot for feature 1.2 showing log-in to account

After submitting their information, if the login is invalid, the login page resets, and users are promptly notified with the error message “Invalid username or password,” as shown in Figure 6. Upon successful login, users are greeted by name and gain access to the navigation bar. Here, they can choose to play a game, access the player dashboard, visit the player store, or sign out.

Invalid username or password.

Figure 6 screenshot for feature 1.2 showing invalid user

Feature 1.3

As shown in Figure 7, Feature 1.3 enables users to log out at any time. Upon logging out, users are redirected to the homepage and lose access to the navigation bar, ensuring a smooth transition for users exiting the platform.



Figure 7 screenshot for feature 1.3 showing log-out of account

Feature 2.1

Figure 8 showcases part of Feature 2.1, where users can choose their preferred language for playing the game by clicking on the corresponding flag icon. Upon selecting a flag, users are directed to a page where they can engage with the worNDly game in their chosen language.



Figure 8 screenshot for feature 2.1 showing select desired language

In Figure 9, another aspect of Feature 2.1 demonstrates the interface subsequent to a user's language selection (in this case, English). Upon choosing a language, the game dynamically selects a word from the corresponding language file to serve as the word of the day. In this particular instance, the selected word is “CHIRR.” Additionally, the interface displays the user’s remaining number of guesses, available games, and the total number of games played.



Figure 9 screenshot for feature 2.1 showing gameplay screen

In Figure 10 below, we witness a scenario where the user submits a word that does not adhere to the worNDly game criteria. Specifically, the input contains numbers, triggering an error message that reads, “Invalid word. Please enter a valid five-letter word.”



Figure 10 screenshot for feature 2.1 showing invalid guess

In Figure 11, we observe an example of an acceptable word submission within Feature 2.1 of the worNDly site. While this word is not the word of the day, the feedback provided indicates the correctness of each letter's placement. Correctly positioned letters are highlighted in dark green, while letters that belong to the word of the day but are in incorrect positions are marked in yellow. Letters that are not part of the word remain unmarked in blue.

Guess a word in English!

Word of the day: CHIRR

Guesses left: 4

Games allowed to play today: 3

You've played: 1

Should be a FIVE letter word.

Enter your guess!

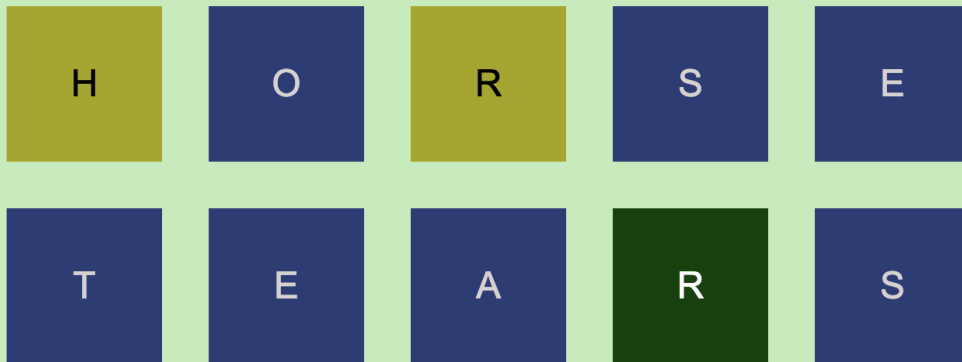


Figure 11 screenshot for feature 2.1 showing valid guess

In Figure 12, we witness an example of a correct guess being submitted to the worNDly site. The word is displayed entirely in green, accompanied by a congratulatory message: “Well done! You won in (number) guesses.” This signifies the end of the game, celebrating the player’s success.



Figure 12 screenshot for feature 2.1 showing correct guess

Feature 3.1

Figure 13 showcases the functionality of Feature 3.1, enabling users to conveniently review their past game plays. On the right side of the screen, a list is presented beneath a dropdown menu containing the user’s previous plays. Users can filter the list by selecting different options from the dropdown menu, such as “This Week.” Upon selection, only plays occurring within the specified time frame will be displayed. The feature provides essential details, including the date of each game, whether the user successfully guessed the word, and the number of attempts taken. This comprehensive overview enhances user experience by facilitating easy access to game history and statistics.

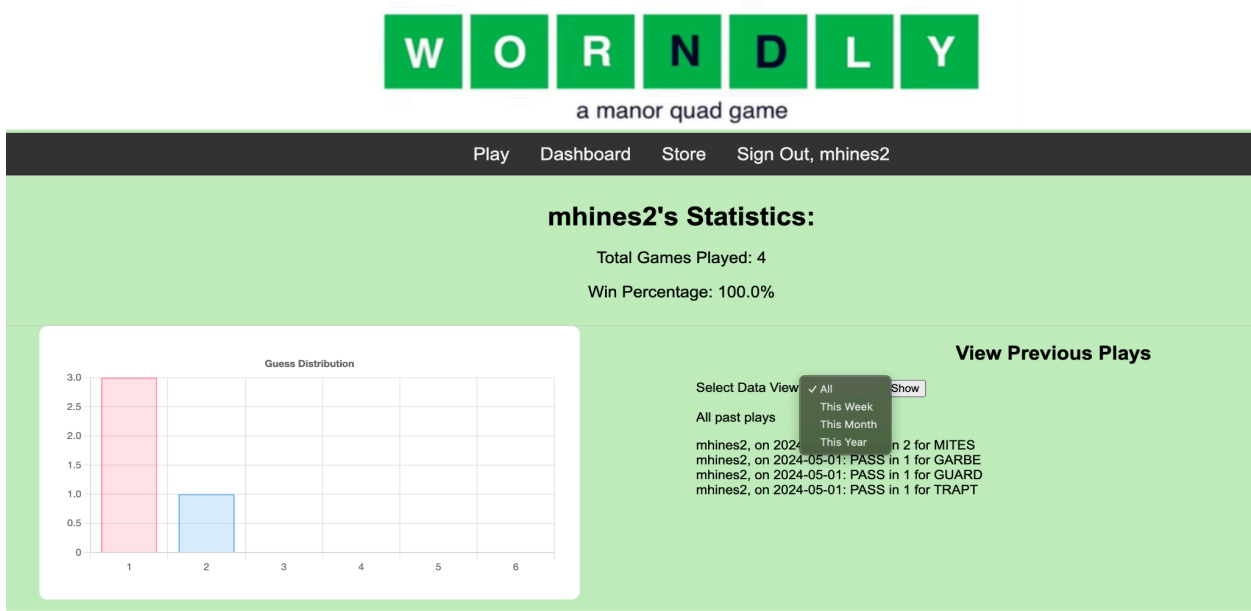


Figure 13 screenshot for feature 3.1 showing all prior plays

Feature 3.2

In Figure 14, we observe the implementation of Feature 3.2, which displays fundamental statistics regarding the user’s past game plays. These statistics include the total number of plays, the frequency of wins, and the distribution of the number of attempts required to guess the correct word. This feature provides users with valuable insights into their gameplay history, allowing them to track their progress and performance over time.

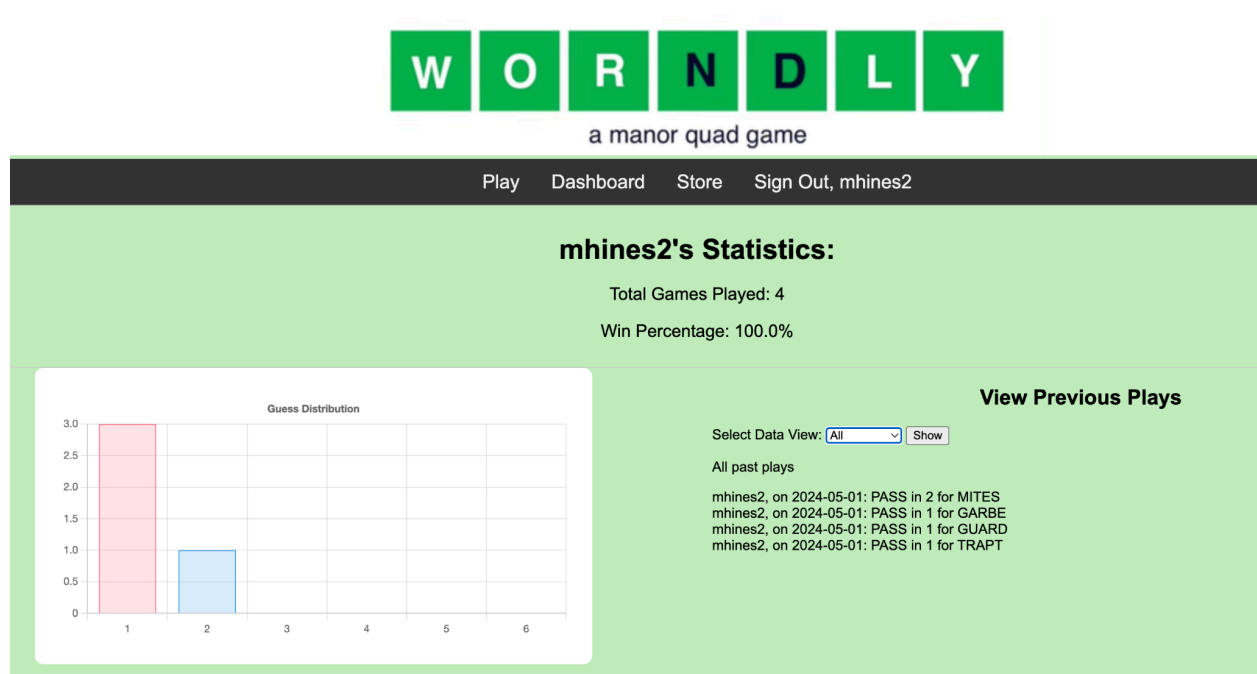


Figure 14 screenshot for feature 3.2 showing basic statistics

Feature 4.1

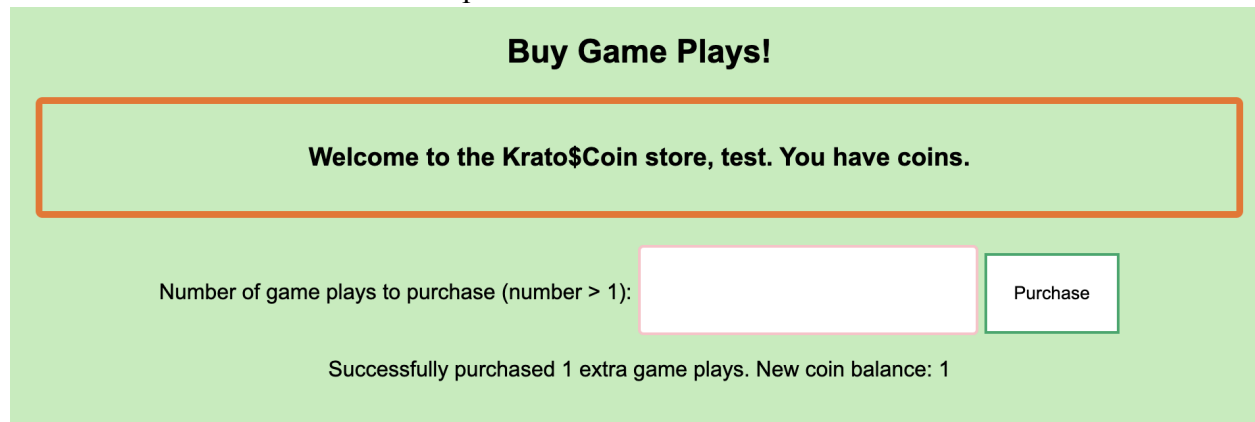
As shown in Figure 15, Feature 4.1 allows users to augment their gaming experience by purchasing additional game plays per day using Krato\$Coins. Priced at a rate of 1 Krato\$Coin per extra gameplay, users are presented with their current coin balance during the checkout process. They specify the desired number of games to purchase, and upon completion, the web application verifies the transaction's success via an external REST API response.

The screenshot shows the 'Buy Game Plays!' interface. A green banner at the top says 'Buy Game Plays!'. Below it, a message box says 'Welcome to the Krato\$Coin store, test. You have 2 coins.' At the bottom, there is a form with a label 'Number of game plays to purchase (number > 1):', a text input field containing '1', and a 'Purchase' button.

Number of game plays to purchase (number > 1):

Figure 15 screenshot for feature 4.1 showing current balance and purchase button

If the transaction succeeds, users receive a confirmation message, indicating the successful acquisition of additional games, as shown in Figure 16. These purchased games accumulate over time and do not expire after 24 hours.



The screenshot shows a web interface titled "Buy Game Plays!". At the top, a green box contains the text "Welcome to the Krato\$Coin store, test. You have coins." Below this, there is a label "Number of game plays to purchase (number > 1):" followed by a text input field. To the right of the input field is a green "Purchase" button. Below the input field, a message states "Successfully purchased 1 extra game plays. New coin balance: 1".

Figure 16 screenshot for feature 4.1 showing purchase of more games

Conversely, if the transaction fails due to insufficient funds or other reasons, users are promptly notified with an appropriate error message “Insufficient funds to purchase game plays,” as shown in Figure 17.



The screenshot shows the same "Buy Game Plays!" interface. The green box at the top still contains "Welcome to the Krato\$Coin store, test. You have coins." Below the label "Number of game plays to purchase (number > 1):", the text input field now contains the number "4". The green "Purchase" button is still present. Below the input field, an error message states "Insufficient funds to purchase game plays".

Figure 17 screenshot for feature 4.1 showing insufficient funds

Project’s Learned Lessons

- 1. What programming paradigm(s) have you chosen to use and why? If you were to start the project from scratch now, would you make different choices? Do you think the paradigm(s) chosen helped (or not) in developing the project?***

For this project, we primarily utilized the object-oriented programming (OOP) paradigm inherent in Python and Django, driven by its suitability for web application development, offering reusable and modular components that facilitate code management and scalability. Django’s built-in support for OOP principles, such as models, views, and templates, provided a structured framework for organizing our code. Additionally, we incorporated aspects of the

functional programming paradigm, particularly in handling data transformations and operations within our views and controllers, leveraging Python's functional capabilities for concise and expressive code. If we were to start the project from scratch, we might emphasize a clearer separation of concerns and adherence to the MVC architecture from the outset to enhance maintainability and scalability. Overall, the chosen paradigms, OOP and functional programming, were instrumental in developing the project, providing a solid foundation for robust, maintainable, and scalable web applications, effectively managing complexity, promoting code reuse, and streamlining development workflows to successfully implement the required features.

2. What were the most intellectually challenging aspects of the project?

One of the most intellectually challenging aspects of the project was navigating the integration of multiple Django apps within a single website. This required a deep understanding of how each team member structured their models and views within their respective apps, as well as thoughtful consideration of the data and functionalities shared between different components. As a result, we had to carefully design our code to ensure clarity and coherence, making it easily understandable for all team members. Additionally, we encountered difficulties with collaboration on GitHub, particularly with merge conflicts arising from simultaneous updates to the same branch. Over time, we improved our communication and coordination strategies, learning to effectively utilize branches and maintain clear communication regarding project changes to mitigate these challenges.

3. What aspects of your process or your group's organization had the largest positive effect on the project's outcome?

One of the most impactful decisions that greatly influenced the outcome of our project was the transition from our custom user model to utilizing Django's built-in user authentication model during Phase 2. Initially, we had developed our own user model for handling login and sign-up functionalities. However, recognizing the benefits of leveraging Django's robust authentication system, we made the strategic choice to integrate it into our project. This shift streamlined our workflow significantly, as we could seamlessly utilize the admin interface and other pre-built Django functionalities. With Django's user authentication, managing user information across different features became much more efficient, allowing us to easily implement features like referential integrity by utilizing usernames as foreign keys. Furthermore, leveraging Django's authentication enhanced the security of our website, enabling us to restrict access to certain components until users successfully signed in. Additionally, integrating Django's user authentication facilitated smoother interaction with the REST API, as we could effortlessly access users' emails when necessary. Overall, this strategic decision significantly optimized our project's functionality, security, and overall user experience.