

CS4530 Final Project: Database Integration

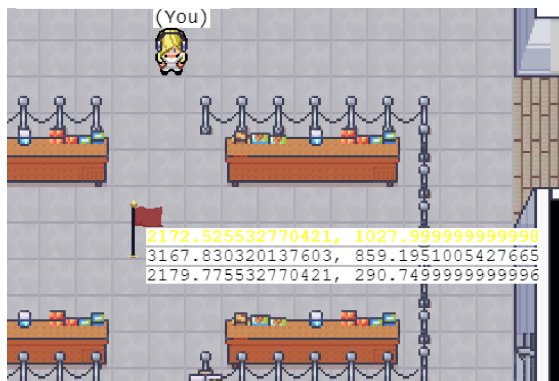
Group 610: Hyoboem Kim, Mario Gonzalez, Dmitry Matveev, Dylan Pham

Our Feature

Our project focuses on introducing persistence to Covey.Town through the implementation of a database that tracks user information, which enhances user functionality. Users can now benefit from secure access through User Authentication and Registration, ensuring privacy and personalization. The addition of User Session Tracking enables users to keep track of the time spent using the app. Moreover, we've implemented a valuable feature allowing the use of flags to mark and save specific locations, providing users with a more organized and navigable virtual environment.

Future Work

Looking ahead, there are several avenues for future enhancements in our project. Firstly, the addition of more features to leverage the relational database structure, such as incorporating a friends list, could further enrich the user experience in Covey.Town. This could enhance social interactions within Covey.Town. Another feature that we wanted to implement was an integrated messaging system, like a global chat and DM functionality. Integrating this with our database would allow users to save messages they like. Finally, we would also expand the functionality of flags to include additional information and actions. For example, users could leave notes or create shared flags for collaborative projects, providing a versatile tool for interaction and organization.



Players In This Town

Current town: test
Dylan Pham

my session count: 2

Active Conversation Areas:
No active conversation areas

Technology Overview

A significant portion of our design considerations centered around creating an extensible database to meet our goals efficiently. The User API design involves the UserController.ts file, where TSOA is utilized to mark functions as GET, DELETE, PATCH, POST, etc. The core operations occur in the UsersService.ts file, isolating endpoint logic and handling queries and error validation. Notably, we refined our design decisions, aligning with industry standards, such as switching from a request body to query parameters for the get User endpoint. Our approach adheres to RESTful API principles, including considerations for naming, status codes, and headers. For the database, we opted for a relational structure to accurately represent relationships, leveraging a one-to-many relationship for users and flags. The relational database design allows for future expansions, like implementing a friends list. Additionally, we adopted a singleton pattern to maintain a single database connection for optimal performance.

Demo and Source

Demo: <https://covey-town-project-610-frontend.onrender.com/>

Source code: <https://github.com/neu-cs4530/fall23-team-project-group-610>

```
model User {
  id Int @id @default(autoincrement())
  flags Flag[]
  messages Message[]
  username String
  email String @unique()
  sessionCount Int @default(0)
}
```

