

Upload Chat Service

Michael Webb
@mjwebb1993 on GitHub

Description

Upload Chat Service, or UCS, serves to offer users simple and easy to use features, such as uploading data and real time communication, in a single piece of software. This will give users an opportunity to streamline their everyday needs into one place, instead of utilizing various pieces of software.



Features

- Creating and Authentication of User Accounts
- Upload various types of data and manage it through CRUD functionality
- Access to a text based chat room for real time communication



Planning - User Stories

- Account Creation User Story

- A new user will be able create an account with confirmation, and then instantly login to the account.

- Chatroom User Story

- A user will be able to create a username and join other users inside the chatroom to communicate in real time.

- FileUpload User Story

- A user will be able to select a file from their harddrive and upload it to a sharable link on the server



Planning - Database

There are two tables in my database, “users” and “files”:

- The users table has 3 columns, “id”, “pw_hash”, and “username”, which is used in the application to authorize users, safely store their passwords, and give users their own identity
- The files table has 3 columns, “id”, “file_name”, and “file_type”, which is used to help identify the data in the db and assign it to a user



Technology Stack

- Java, Javascript
- Spring Boot
- ThymeLeaf
- MySQL
- Gradle, WebSocket, Bootstrap, JQuery, Hibernate



Demo



What I Learned

- How to implement a WebSocket into my application
- How to utilize existing utilities in the Spring Framework to upload data through the application to an already established database
- The importance of resource management in structuring the project and how to better organize various controllers



What's Next

- Improving the chat feature so a user no longer has to reenter a username to enter the chat room
- Truly connecting the uploaded files to specific users in the database and being able to access the files through the database and not just a link
- Creating a more uniform UI throughout the app, some pages also have a primitive look that needs updating for a more positive user experience
- Learning how to be more efficient with WebSockets and potentially adding a saved history feature to the chat room

