Fitz: Phase 2 Report

Creators: Wesley Vance Blair, Katarina Marsteller

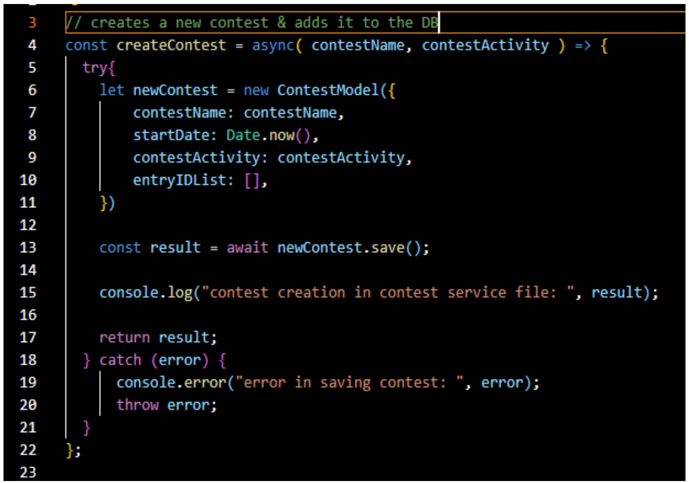
Implementations

We were able to implement the commenting system for the user posts and the contest setup. The user is capable of adding comments to a post and communicating with the database.

For the contests, we were able to implement the CRUD features needed for them. Firstly, the creation of a new contest uses the createContest function present in ContestService(Fig 1). Furthermore, the creation of the contest adheres to the model created in ContestModel (Fig2). Finally, the program is capable of gathering all contests within the database, updating a contests' activity, and removing a contest from the database.

In addition, users now have the ability to add comments to posts. Adding comments utilizes a new function called addComment found in PostService.js. This allows for a new comment to be added to a list contained within the PostSchema(Fig 3). Moreover, within the PostModel, a comment schema is established to allow information about a comment to be captured.

With time, we would be able to implement and polish some of the functionality to achieve our idea for Fitz in full.



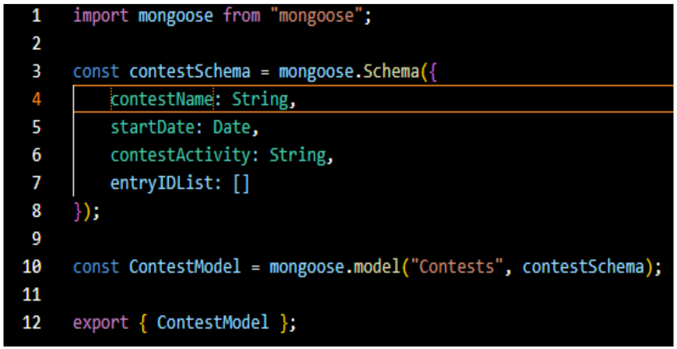
```
// creates a new contest & adds it to the DB
const createContest = async( contestName, contestActivity ) => {
  try{
    let newContest = new ContestModel({
        contestName: contestName,
        startDate: Date.now(),
        contestActivity: contestActivity,
        entryIDList: [],
    })

    const result = await newContest.save();

    console.log("contest creation in contest service file: ", result);

    return result;
  } catch (error) {
     console.error("error in saving contest: ", error);
     throw error;
  }
};
```

Fig. 1 - createContest function



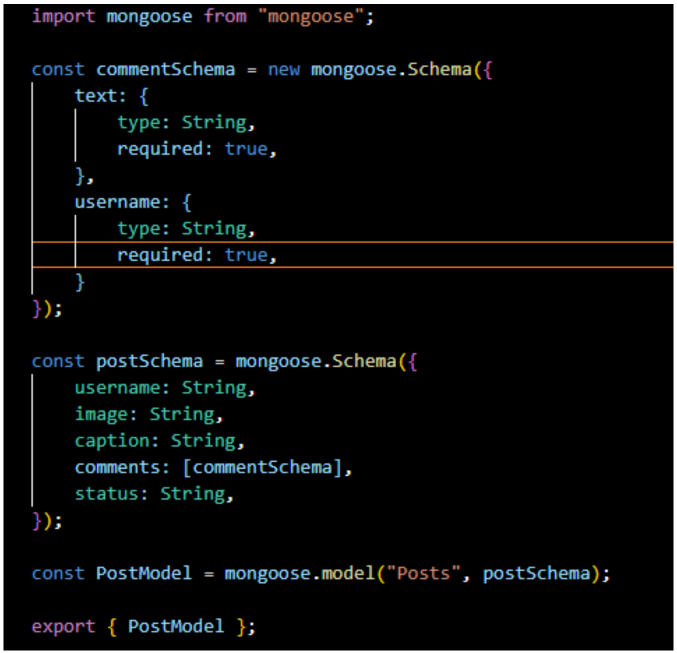
```
import mongoose from "mongoose";

const contestSchema = mongoose.Schema({
    contestName: String,
    startDate: Date,
    contestActivity: String,
    entryIDList: []
});

const ContestModel = mongoose.model("Contests", contestSchema);

export { ContestModel };
```

Fig. 2 - Contest model



```
import mongoose from "mongoose";

const commentSchema = new mongoose.Schema({
    text: {
        type: String,
        required: true,
    },
    username: {
        type: String,
        required: true,
    }
});

const postSchema = mongoose.Schema({
    username: String,
    image: String,
    caption: String,
    comments: [commentSchema],
    status: String,
});

const PostModel = mongoose.model("Posts", postSchema);

export { PostModel };
```

Fig. 3 - PostModel.js schema