**Project 4 Planning**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project 4 - Project Plan** | | | | |
| **Planning Phase** | | |  |  |  |
|  | Create a planning directory | |  |  |  |
|  | ✔️ | This project plan document |  |  |  |
|  | ✔️ | Create user stories (features, requirements of this application) |  |  |  |
|  | ✔️ | Create Wireframe images |  |  |  |
|  | ✔️ | Create a Github Repo |  |  |  |
| **Development Phase** | | |  |  |  |
|  | Work on the Express Backend application | |  |  |  |
|  | ✔️ | Pseudocode before you write the code |  |  |  |
|  | ✔️ | Review the Muse app reviewed in class | <https://git.generalassemb.ly/jdr-0622/muse-express-api-backend-lesson> | |
|  | ✔️ | On config.json, change database to "project2\_development" |  |  |  |
|  | ✔️ | Route to get all trivia questions |  |  |  |
|  | ✔️ | Route to add a new trivia question |  |  |  |
|  | ✔️ | Route to display one trivia question |  |  |  |
|  | ✔️ | Route to delete a question |  |  |  |
|  | ✔️ | Route to edit a question |  |  |  |
|  | ✔️ | Route to update a question |  |  |  |
|  | ✔️ | Route to get all categories |  |  |  |
|  | ✔️ | Route to add a new category |  |  |  |
|  | X | Route to get one specific category |  |  |  |
|  | X | Route to edit a category |  |  |  |
|  | X | Route to update a category |  |  |  |
|  | X | Route to delete a category |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  | React Frontend Application | |  |  |  |
|  | ✔️ | Displays all the trivia questions |  |  |  |
|  | ✔️ | Displays the trivia detail when you click on a trivia question. |  |  |  |
|  | ✔️ | Displays the trivia question category |  |  |  |
|  | ✔️ | Allows adding a new trivia question | Problem |  |  |
|  | ✔️ | Allows updating a trivia question |  |  |  |
|  | ✔️ | Allows deleting a trivia question |  |  |  |
|  | ✔️ | Trivia Detail CSS |  |  |  |
|  | ✔️ | \* Dark blue background, white text |  |  |  |
|  | ✔️ | \* Spacing and font looks correct |  |  |  |
|  | ✔️ |  |  |  |  |
|  | ✔️ | Displays the Category page |  |  |  |
|  | x | Allows adding new category |  |  |  |
|  | x | Allows deleting a category |  |  |  |
|  | x | Allows updating a category |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **Project Submission** | | |  |  |  |
|  | Requirements | |  |  |  |
|  |  |  |  |  |  |
|  |  | Post a link to the backend Repo |  |  |  |
|  |  | - Readme file with adequate documentation |  |  |  |
|  |  | Post a link to the frontend Repo |  |  |  |
|  |  | Post a link to the deployed backend application |  |  |  |
|  |  | post a link to the deployed frontend application |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

<https://git.generalassemb.ly/jdr-0622/project-4>

<https://github.com/marcwright/deere-project4-express-api-starter>

You can fork and clone this ^ code to get started. **Done**

**Issues/Questions**

1. When I start up my app I get message Error: Invariant failed: You should not use <Link> outside a <Router>. **FIXED**
2. On my allTrivia screen, I have a form to add a new trivia question. I followed the model in the Muse app, but I’m not sure its right. **Fixed**
3. When I set up my triviaQuestion array, it needs to hold the question and the 4 answers. Is it defined correctly? **YES**
4. My input fields to edit an existing record don’t allow the data to be edited. **Fixed**
5. addTrivia is not being called when user clicks on “Add Trivia Question” button. **Fixed**
6. How do I handle it if the frontend app passes bad or incomplete data to the backend app to add a new record?
7. After I delete a record, I want the program to automatically return to the allTrivia screen. Also, the record I just deleted still shows up on the all Trivia screen until I hit refresh. **Fixed**
8. My update method is not being called at all. **Fixed** (did not pass the updateTrivia method in the route)
9. After adding a new record, when I go to the Trivia Detail page, the program errors because the trivia data and corresponding category are not saved in the array in the program. **Fixed**
10. After I add a trivia record, if I click on that record to look at it on the detail page, the program dies. **Fixed**
11. Need to test my edit/update functions. **Done**
12. Trivia Categories list needs to be view only, not clickable.  **Done**

**Necessary Deliverables**

Your submission must include **all** of the following:

**Project Idea and Initial Planning**

By 11:30am CST on Thursday, September 10, 2020, you need to submit:

1. Your project idea (a brief 2-3 sentence description of your app)
2. A list of your models and their properties

These and any other planning assets (wireframes, user stories, etc) should go inside a planning/ directory inside one of your project repositories. Planning and Process is one category by which we will be [evaluating your project](https://git.generalassemb.ly/jdr-0622/project-4/blob/master/evaluation.md).

The repository/repositories for your final project should include:

* A working project (built by you) that meets the technical requirements above.
* Frequent commits dating back to the very beginning of the project.
* A readme.md file in the root of your repo that follows [good standards of documentation](https://git.generalassemb.ly/jdr-0622/documentation-markdown) (i.e. explanations of the technologies used, the approach taken, features, installation instructions, unsolved problems, etc.)
* A planning/ directory in the root of your repo containing a diagram mapping out your project domain. You are welcome to include other planning documents (e.g., wireframes, user stories).
* A link to your hosted, working project in the URL section of your Github repo:

### Pre-deployment checklist

* Add the deployed URL to repo settings
* Add a readme with:
  + An embedded screenshot of the app
  + Explanations of the **technologies** used
  + **Installation instructions** for any dependencies
  + Link to your **user stories** – who are your users, what do they want and why?
  + Link to your **wireframes** – sketches of major views / interfaces in your application
  + Descriptions of any **unsolved problems** or **major hurdles** you had to overcome

**Ideas**

1. Trivia database
   1. Express backend that stores trivia questions and answers in a Postgres database
   2. React frontend that uses the routes in the express backend to view, add, update and delete records in the database.
   3. Category dropdown. Allow user to select a category from a dropdown box.
   4. A category can have many trivia questions. A trivia question can belong to only one category.
   5. Stretch goals
      1. Have a field that gives the difficulty level of each question.
      2. Search feature. Allow user to specify a search criteria to find trivia questions where the question or the answer contains a specific keyword.
      3. Have a backend that adds, updates or deletes trivia categories.
      4. Have a link to the trivia game from project 1 and have it use the trivia database in project 4.