Star Wars Trivia

Cohort #

### Table of Contents

[Table of Contents](#_7vbf9ja7xyoi)

[Mission Statement](#_j2qiseiwi40z)

[User Stories](#_gck1okzdy64z)

[User 1: The guest](#_vtxdq1oay2wk)

[Database](#_4j4fk6v9vrfo)

[Tables](#_pc03rh351huh)

[Endpoints](#_sr61o23qri28)

[Features](#_pamvuvex3ump)

[Wireframe](#_q85md16hzpog)

[Schedule](#_ye3wz1kg9ovv)

[Final Notes](#_9561h8mlbskk)

### 

### 

### Mission Statement

This project is a fun trivia app for Star Wars enthusiast!

### 

### User Stories

This is where you can work on developing a collection of user stories. These user stories should be planned through the perspective of your expected audience and each of the potential roles your end user may have.

Here’s an example:

#### *User 1: The guest*

*As a guest, a user can use the app to test their Star Wars knowledge with family and friends.*

### 

### Database

This is where you write out what your database will look like. List out each table, the columns (include the dataTypes), and the database associations your project will have on the server-side.

#### Tables

Here’s an example:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Users*   |  |  |  |  |  | | --- | --- | --- | --- | --- | | *ID* | *FirstName* | *LastName* | *Email* | *Password* | | *num* | *string* | *string* | *string* | *string* | |

Table 1

Bounty Hunter

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Galactic Era info | Rebellion era info | Resistance era info | Response | bountyId |  |
| String | Num | Num | Num | \*HELP\* | num |  |

Table 2

Response table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| responseId | questionId | questionChoiceId | isRight |  |  |  |
| num | num | num | bool |  |  |  |

Table 3

Question table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| questionId | Question | isActive | era |  |  |
| Num | Object, stirng?? | bool | Enum(GE,RE,RESE) |  |  |

Additional Tables

Question\_choice table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| questionChoiceId | questionId | isRightChoice |  |  |  |
| Num | Num | bool |  |  |  |

#### Endpoints

Here’s an example of a collection of endpoints for a single table:

*Users: ~/api/users*

*GET/userInfo/:id => Get user by ID*

*POST /register => Registers new user account*

*POST /login => Logs in a user*

### Features

Features are instances or examples of different pieces of functionality. This is where you list out the features you are planning on implementing. Consider the different steps and logic those features required to do the expected job. This could include fetching data from a 3rd party API or simply looping over data from your server. Differentiate between your version 1.0 or MVP (minimal viable product) and version 2.0 or stretch goals.

### 

|  |  |
| --- | --- |
| Version 1.0 / MVP | Version 2.0 / Stretch Goals |
| * *Users can create accounts* * Get questions from different eras * Get 2-4 possible answers to choose from * Create a bounty hunter * Get points for answering questions * Leveling up their bounty hunter for each era | * *Users can reset passwords* * Throw in other questions like (star trek) |

### 

Data Flow

When planning a project, we also need to consider what data needs to be available in certain components. Where do you need to declare your variable and which components feed into each other so that the value of the variable is accessible when you click here or navigate to this part of your project? Building out the data flow of your project will also be helpful when building out your folder structure. Take the time to draw out the structure of your project, where you will declare certain variables and build out pieces of logic, and trace the flow of where they will be utilized.

### Wireframe

This is where you draw out a blueprint or a skeleton of the visual layout of the project. This is also known as the UI (User Interface) of a project. This can be drawn out with paper and pencil, whiteboards, and/or tools such as [sketch](https://www.sketch.com/), [xd adobe](https://www.adobe.com/products/xd.html), or [balsamiq](https://balsamiq.com/). Consider how the components will fit into each and what data needs to be passed between components. This is also a great place to think about adding a [style guide](https://speckyboy.com/inspirational-examples-ui-style-guides/) and words or images that you are using for inspiration.

### 

### Schedule

This project is the equivalent of one sprint in the agile methodology. In this section, write out a schedule spanning over the next couple of weeks. This should include deployment, time set aside to tackle especially challenging features, testing, etc. The table below is a guide. It is not necessary to specify each day’s work/logic. Feel free to estimate your time. Consider whether you will be working over the weekend(s).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** | **Day 6** | **Day 7** |
|  |  |  |  |  |  |  |
| **Day 8** | **Day 9** | **Day 10** | **Day 11** | **Day 12** | **Day 13** | **Day 14** |
|  |  |  |  |  |  |  |
| **Day 15** | **Day 16** | **Day 17** | **Day 18** |  |  |  |
|  |  |  |  |  |  |  |

### Final Notes

Great job with planning! You are now set to start coding. Planning a project is incredibly beneficial to your success and the success of your project. Here are some resources to help you with your planning.

* [How to plan a web application](https://selftaughtcoders.com/plan-web-application/)
* [Step By Step: Planning a web application](https://medium.com/@ericwindmill/step-by-step-planning-a-web-application-ddaa010a8353)