Lab 6

Submit Assignment

Due Thursday by 11:55pm **Points** 100 **Submitting** a file upload **File Types** zip

CS-546 Lab 6

JSON Routes

For this lab, you will create a simple server that will provide data similar to lab 5, but this time from a server.

For this lab, you will not need to use a database. You can store your data right in your routes.

Your routes

`/about`

When making a GET request to `http://localhost:3000/about`, this route will return JSON in the following structure (with your own information):

```
{
  "name": "Your Name",
  "cwid": "Your CWID",
  "biography": "2 biography paragraphs separated by a new line character (\n).",
  "favoriteShows": ["array", "of", "favorite", "shows"],
  "hobbies": ["array", "of", "hobbies"]
}
```

`/story`

When making a GET request to `http://localhost:3000/story`, this route will return the following JSON:

```
{
  "storyTitle": "Story Title",
  "story": "Your story.\nSimply use line breaks for paragraphs.\nLike this."
}
```

`/education`

When making a GET request to `http://localhost:3000/education`, this route will will return JSON in the following structure (with your own information):

```
[
```

```
    {
      "schoolName": "First School Name",
      "degree": "First School Degree",
      "favoriteClass": "Favorite class in school",
      "favoriteMemory": "A memorable memory from your time in that school"
    }
]
```

Make sure to include at **least** 3 schools.

Packages you will use:

You will use the **express** package as your server.

You can read up on **express** (http://expressjs.com/) on its home page. Specifically, you may find the **API Guide section on requests** (http://expressjs.com/en/4x/api.html#req) useful.

You may use the **lecture 6 code** (https://github.com/Stevens-CS546/CS-546/tree/master/Lecture%20Code/lecture_06) as a guide.

You must save all dependencies to your package.json file

Requirements

1. You **must not submit** your node_modules folder
2. You **must remember** to save your dependencies to your package.json folder
3. You must do basic error checking in each function
    1. Check for arguments existing and of proper type.
    2. Throw if anything is out of bounds (ie, trying to perform an incalculable math operation or accessing data that does not exist)
    3. If a function should return a promise, instead of throwing you should return a rejected promise.
4. You **must remember** to update your package.json file to set `app.js` as your starting script!
5. You **must** submit a zip, rar, tar.gz, or .7z archive or you will lose points, named in the followign format: `LastName_FirstName_CS546_SECTION.zip` (or, whatever the file extension may be). You will lose points for not submitting an archive.