Lab 6

Submit Assignment

Due Thursday by 11:59pm **Points** 100 **Submitting** a file upload

CS-546 Lab 6

JSON Routes

For this lab, you will create a simple server that will represent the same data that you created in lab 5 by sending JSON down through API calls.

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

Your routes

`/about`

This route will return the following JSON:

```
{
  "name": "Your Name",
  "biography": "2 biography paragraphs seperated",
  "favoriteShows": ["array", "of", "favorite", "shows"],
  "hobbies": ["array", "of", "hobbies"]
}
```

`/story`

This route will return the following JSON:

```
{
  "storyTitle": "Story Title",
  "story": "Your story"
}
```

`/education`

This route will return the following JSON:

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

```
    }
]
```

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 4 code** (https://github.com/Stevens-CS546/CS-546-WS-Summer-1/tree/master/Lecture%20Code/lecture_4) as a guide.

You may use the **lecture 5 code** (https://github.com/Stevens-CS546/CS-546-WS-Summer-1/tree/master/Lecture%20Code/lecture_5) as a guide.

You may use the **lecture 6 code** (https://github.com/Stevens-CS546/CS-546-WS-Summer-1/tree/master/Lecture%20Code/lecture_6) 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.