**Roadblocks**

* **Time Management:** Since I am working a full-time job, many of the ideas that I had were pushed back, in favour of getting a working product out by the due date. This was especially muddled by how I was unfamiliar with the technologies, making it more difficult to judge how long tasks were going to take.
* **Learning New Technologies:**

**Refactoring Opportunities**

* **[Back-end]** Add a **User** entity/object and scaffold a controller for it, to allow proper HTTP client methods (**PUT, POST, GET, DELETE**) for that entity/object.
  + **Issues with Current Code:**
    1. Currently not coded in a way where multiple users can use the application, even within an In-Memory Database
    2. Storing user information (ie. **userAnswers**) in the front-end, when it should be in the back-end
  + **How Solution will Improve Current Code:**
    1. If coded to include a **User** entity/object, then it will be easier to refactor in a future iteration to add functionality for multiple users
    2. Increased separation of roles between the front- and back-end – the front-end will be used for mainly view-related roles, while the back-end will primarily be used for data storage and manipulation
* **~~[Back-end & Front-end]~~** ~~Remove the~~ **~~selectedAnswer~~** ~~and~~ **~~isSelected~~** ~~fields in the~~ **~~Question~~** ~~entity/object.~~
  + **~~Issues with Current Code:~~**
    1. ~~Contains extra fields/properties within the~~ **~~Question~~** ~~entity/object that aren’t being used.~~
  + **~~How Solution with Improve Current Code:~~**
    1. ~~Removes extra fields that aren’t being used, keeping the code less cluttered.~~
* **[Testing]** Automate testing using Selenium.
  + **Issues with Current Method:**
* **~~[Back-end & Front-end]~~** ~~Refactor so that the database references an~~ **~~AnswerSet~~** ~~DB that has all the answer sets.~~
* **~~[Front-end]~~** ~~Refactor back and next questions into one method; there is no need to have two if similar code is used (and use that one method for the Finish button too?)~~
  + ~~Pass variable to indicate whether Back or Next button called it to know what needs to be done; otherwise, subtract or add 1 from the HTML file and pass that calculation over~~
* **[Front-end]** Style refactoring:
  + Change the buttons to look more sophisticated
  + Add icons?
  + Add sliding animations for the questions, and the same animation as the start card for the suggestion card
  + Change the start and finish buttons to look more sophisticated
  + Refactor to put common styles in the global styles.css
* **[Back-end & Front-end]** Clear out all unnecessary objects or code, such as base-panel, mock-dbs, etc. Weather ForecastController, etc