Group Project – Developing Apps Using Emerging Web Technologies

Due Date: Group presentation in Week 14.

Purpose: The purpose of this project is to:

- Design and code web apps using emerging Web Frameworks
- Build a Rest or Graph QL API using Express, Next.js, or Gatsby
- Build a Front-End (React, Svelte, or Angular) that utilizes the Rest/Graph QL API
- Apply appropriate design patterns and principles
- Use Machine Learning to make intelligent use of data

References: Read the textbook, lecture slides, class examples, and additional references provided here. This material provides the necessary information that you need to complete the project. You may need to read and use more materials and tools to implement a good solution.

Be sure to read the following general instructions carefully:

- This Project **may be completed in groups of 3-4 students**.
- This project can be **replaced with your capstone project** (COMP-231 or COMP-313), if you use and implement the **same front-end/back-end technologies shown in this document**.
- You will have to **present and demonstrate your solution in Week 13 or Week 14** and upload the solution on eCentennial through the assignment link on D2L. Bonus marks will be given if you also **publish the app on Microsoft Azure Cloud, Amazon, or another Cloud platform.**
- **You VS Code project name should be named "YourGroupNameCOMP308Project" and should be zipped in a file YourGroupNameCOMP308Project.zip**.

Project Specifications

Your client needs an application to help nurse practitioners to monitor patients during the first weeks of their release from the hospital and also help the patients to monitor their daily activities. Develop a web app that implements the following functionalities:

1. User **registration/login**

2. If the user is a **nurse**:
    a. Allow the user to enter vital signs: *body temperature*, *heart rate*, *blood pressure*, or *respiratory rate*.
    b. Allow the user to access information captured during a previous clinical visit, vital signs: *body temperature, heart rate*, *blood pressure*, or *respiratory rate.*
    c. Allow the user to send daily motivational tips to the patient (by storing them in the database and providing a daily tips page for the patient to view, etc.).

3. If the user is a **patient**:
    a. Allow the user to create and send an emergency alert to first responders (by storing this in a separate collection)
    b. Allow the user to run motivational video or games that encourage their physical exercises at home. Gaming students are encouraged to design/incorporate their own games/interactive pages.

c. Allow the user to enter daily information as specified by the nurse practitioner (for example *pulse rate*, *blood pressure*, *weight*, *temperature*, *respiratory rate*).
d. Allow the user to use a **checklist of common signs and symptoms**, to generate a list of possible medical conditions and advice on when to consult a healthcare provider.

4. Use **MongoDB** for storing the information.
5. **Use Express for Rest or Graph QL API frameworks. Alternatively, use Next.js or Gatsby.**
   a. **Some choices for front-end frameworks:**
      i. **React** 16.8 or higher
      ii. **Svelte** (https://v2.svelte.dev/)
      iii. **Angular** 7 or higher or **Vue**

Apply **MVC** for the Express part and **responsive** web design principles. Use CSS to create a nice look and feel of your app. Display the logo for the application, other images, game objects, etc.

(100 marks)

Evaluation of software solution (all items need to be shown during the group presentation):

Evaluation Component	Percentage
Functionality:	
Correct MongoDB database (proper use of document structure)	10%
Correct Rest API MVC or Graph QL (proper use of design patterns)	35%
Correct Front End (proper use of architecture/libraries/frameworks)	30%
Friendliness	10%
Innovation (intelligent use of symptoms or other data using machine learning)	15%
Total	**100%**

References:

https://docs.mongodb.com/manual/data-modeling/
https://expressjs.com/en/4x/api.html
http://mongoosejs.com/docs/guide.html
https://reactjs.org/
https://nextjs.org/
https://www.gatsbyjs.org/
https://angular.io/docs
https://vuejs.org/
https://js.tensorflow.org/
https://docs.microsoft.com/en-us/visualstudio/javascript/tutorial-nodejs?view=vs-2017