COVID-19 Virtual Test

Student: Victoria Fischer
Faculty Mentor: Professor Mat Gargano
Department: Computer Science

Abstract

We created an application that displays a user's likelihood of having contracted the virus COVID-19. Using the React framework with some basic CSS and Bootstrap, the application prompts a user with various Coronavirus symptom-related questions that they must answer to the best of their ability. Comparing user's answers with a scoresheet, our custom algorithm will calculate the user a "score". Depending on how consistent this "score" is to our algorithm, they will display three outcomes. "Low-risk, "Medium-risk, or "High-risk". Our algorithm is built with intent to learn and create a feedback loop, helping to further train the AI. The main goal is to provide a way for people to see how their symptoms align with Coronavirus symptoms, and depending on their risk factor, to provide information that best fit the score which they received. This application is in no way meant to be a means of diagnosis or treatment, nor is it meant to be comparable to medical professional's or the CDC's guidelines.

Algorithm

The custom algorithm consists of three states: high risk, low risk, and no risk. Each question in our array of questions has two possible answers- yes or no. "Yes" will indicate the score must add some value, and "No" will indicate no increment of score. Each time the user answers a question with "Yes", the algorithm determines if the question implies a high risk or low risk, choosing to either increment the score by 1 or 3. Each time the user answers a question with "No", a no risk state is sent to our score sheet function and doesn't add any value to the user's score.

```
const questions = [
    {
        questionText: 'Have you lost your sense of taste and/or smell?',
        answerOptions: [
            { answerText: 'Yes', highRisk: true, lowRisk: false },
            { answerText: 'No', noRisk: true},
        ],
        questionValue: 3,
    },
    {
        questionText: 'Have you experienced fever or chills?',
        answerOptions: [
            { answerText: 'Yes', lowRisk: true, highRisk: false },
            { answerText: 'No', noRisk: true},
        ],
        questionValue: 1,
    },
```

Figure 2: Question array

```
    questionText: 'Have you been exposed to someone with COVID-19 in the last two weeks?',
    answerOptions: [
        { answerText: 'Yes', highRisk: true, lowRisk: false },
        { answerText: 'No', noRisk: true },
    ],
    questionValue: 3,

},
{
    questionText: 'Have you been experiencing a sore throat?',
    answerOptions: [
        { answerText: 'Yes', lowRisk: true, highRisk: false },
        { answerText: 'No', noRisk: true },
    ],
    questionValue: 1,
},
```

Figure 3: Question array (continued)

```
const handleAnswerOptionClick = (highRisk, lowRisk) => {

    if (highRisk) {
        setScore(score + 3);
    }
    else if (lowRisk) {
        setScore(score + 1);
    }

    const nextQuestion = currentQuestion + 1;
    if (nextQuestion < questions.length) {
        setCurrentQuestion(nextQuestion);
    } else {
        setShowScore(true);
    }
};
```

Figure 1: Score function

Software

This application is built using the React framework. The React JavaScript library has many components used in this application, such as ScrollToView, useState, BrowserRouter, and much more. We use useState to keep track of the user's score and the current question. We use BrowserRouter to be able to navigate between webpages. Yarn is also installed in the application for the ScrollToView component. ScrollToView makes navigating the application much easier for the user.
Basic HTML and CSS are also used to style the application.

Future Work

Linking a database to the application would be beneficial for a number of reasons:

- Having users be able to make accounts to sign in and out of. If we have a database and can store user information, authenticate users, and store test history, then users can go back and look at their previous test results any time they would like.
- Storing the questions in the question array would make it much easier to modify existing or add new questions to the application. Without the database, we have to add new elements to the array each time. With a database, all we would have to do is input the new question variables.

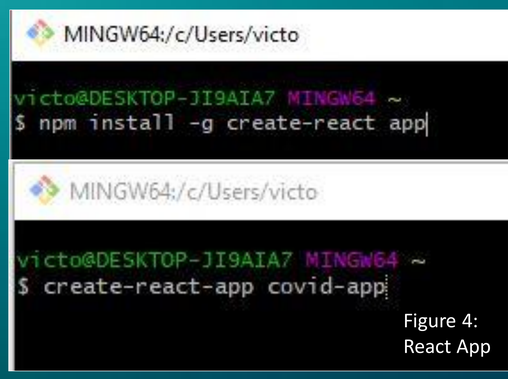

Figure 4: React App