Quaefacta Health
EMPOWER. SECURE. CONNECT.
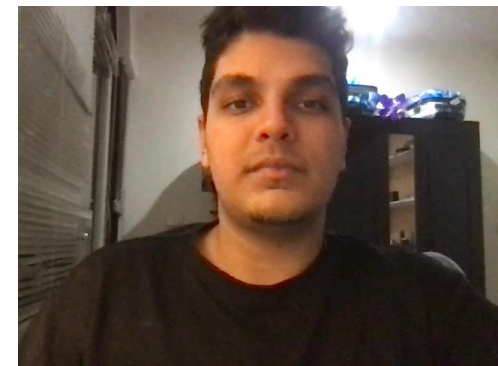
Capstone SS-184
Quaefacta Digital Health Wallet

Richard Forsey, s3857811

Aman Khan, s3811422

Jiajie Lu, s3779102

Jack Allen, s3832293

Quoc Tran, s3827826

Project Goals

▶ Rebuild an app from the ground up.

▶ In this rebuilding, write high quality code that is reliable, extendable and elegant.

▶ Implement features in the full stack.

▶ Implement DevOps to improve collaboration and automate deployment

Implementation
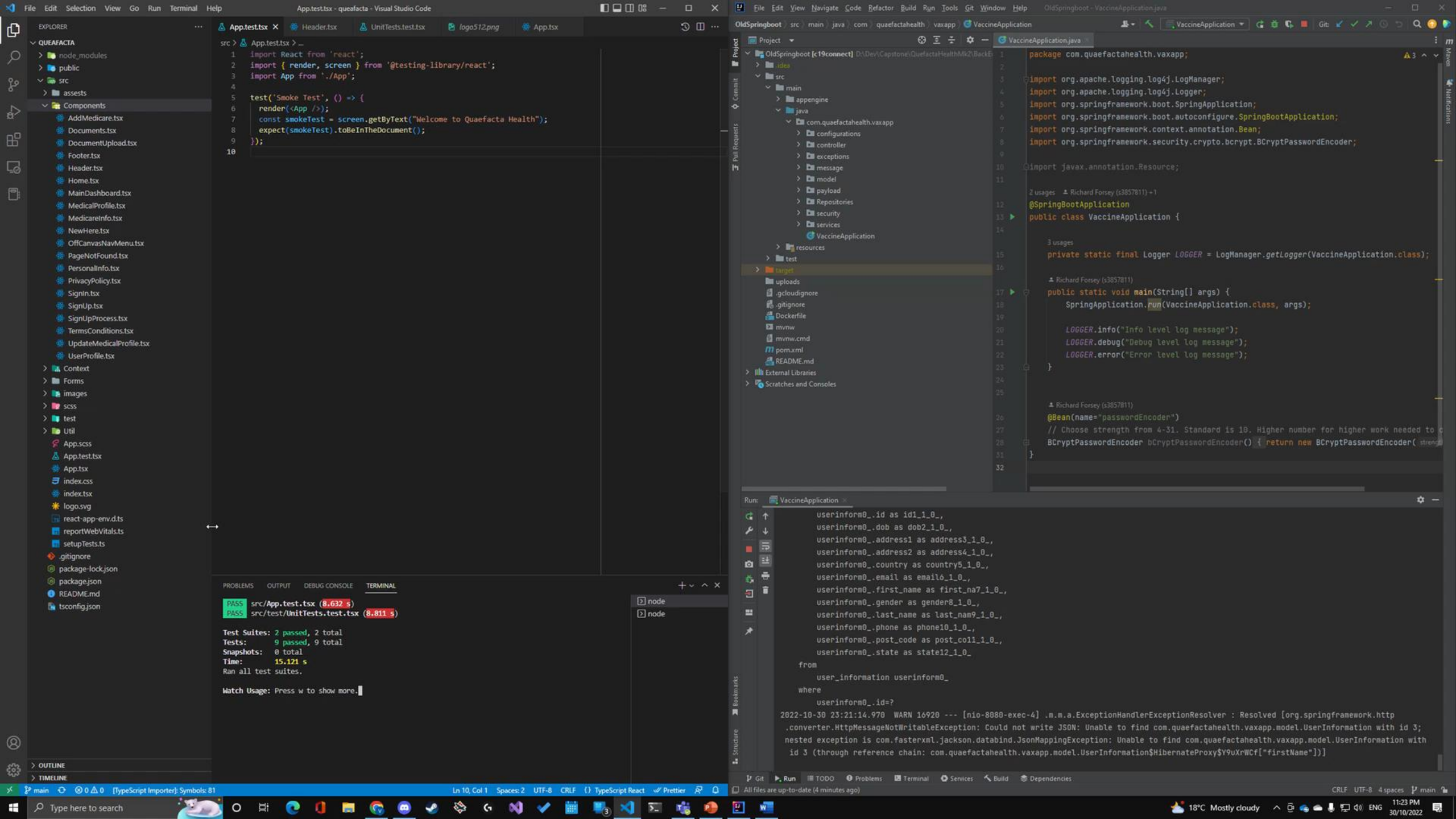
- Frontend (React)
  - Reusable components
  - Bootstrap CSS + custom CSS

- Backend (Java Springboot)
  - MySQL database
  - Springboot ORM for table creation
  - RDS database
  - S3 bucket for file storage

- AWS (Ansible)
  - Scripted infrastructure creation of AWS environment
  - Scripted tear down of AWS environment

Demonstration

- Home

- Sign in

- Sign up

- Information forms

```tsx
import React from 'react';
import { render, screen } from '@testing-library/react';
import App from './App';

test('Smoke Test', () => {
  render(<App />);
  const smokeTest = screen.getByText("Welcome to Quaefacta Health");
  expect(smokeTest).toBeInTheDocument();
});
```

```java
package com.quaefactahealth.vaxapp;

import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import javax.annotation.Resource;

@SpringBootApplication
public class VaccineApplication {

    private static final Logger LOGGER = LogManager.getLogger(VaccineApplication.class);

    public static void main(String[] args) {
        SpringApplication.run(VaccineApplication.class, args);

        LOGGER.info("Info level log message");
        LOGGER.debug("Debug level log message");
        LOGGER.error("Error level log message");
    }

    @Bean(name="passwordEncoder")
    // Choose strength from 4-31. Standard is 10. Higher number for higher work needed to c
    BCryptPasswordEncoder bCryptPasswordEncoder() { return new BCryptPasswordEncoder( strengt
}
```

Terminal output (VS Code):

```
PASS  src/App.test.tsx (8.632 s)
PASS  src/test/UnitTests.test.tsx (8.811 s)

Test Suites: 2 passed, 2 total
Tests:       9 passed, 9 total
Snapshots:   0 total
Time:        15.121 s
Ran all test suites.

Watch Usage: Press w to show more.
```

Run output (IntelliJ):

```
        userinform0_.id as id1_1_0_,
        userinform0_.dob as dob2_1_0_,
        userinform0_.address1 as address3_1_0_,
        userinform0_.address2 as address4_1_0_,
        userinform0_.country as country5_1_0_,
        userinform0_.email as email6_1_0_,
        userinform0_.first_name as first_na7_1_0_,
        userinform0_.gender as gender8_1_0_,
        userinform0_.last_name as last_nam9_1_0_,
        userinform0_.phone as phone10_1_0_,
        userinform0_.post_code as post_co11_1_0_,
        userinform0_.state as state12_1_0
    from
        user_information userinform0_
    where
        userinform0_.id=?
2022-10-30 23:21:14.970  WARN 16920 --- [nio-8080-exec-4] .m.m.a.ExceptionHandlerExceptionResolver : Resolved [org.springframework.http
.converter.HttpMessageNotWritableException: Could not write JSON: Unable to find com.quaefactahealth.vaxapp.model.UserInformation with id 3;
nested exception is com.fasterxml.jackson.databind.JsonMappingException: Unable to find com.quaefactahealth.vaxapp.model.UserInformation with
id 3 (through reference chain: com.quaefactahealth.vaxapp.model.UserInformation$HibernateProxy$Y9uXrWCf["firstName"])]
```

Evaluation

- Good Communication

- Good cooperation

- Actively connect with customers

- A clear division of responsibilities

Highlights

- PROBLEM SOLVING
- CREATIVE THINKING
- CRITICAL THINKING
- TEAMWORK
- NEGOTIATIONS WITH THE CLIENT