Assignment 3: Back-End Development

Group 20

Answer these Questions

Provide the link to your GitHub repository for TAs to view the code. Code should include unit tests.

https://github.com/jenniferceja/COSC4353-Software-Design/tree/main/mern-app/backend
Note: The Volunteer History module got implemented through the profileController and therefore tested through profile's unit test due to our team's initial data object designs. Looking ahead at assignment 4, we see we are supposed to have a separate object dedicated to volunteer history in the DB and will separate it back out.

List what backend technologies you are using and why.

- Node.js - fast, non-blocking runtime, extends our use of Javascript into backend.
- Express.js - simplified routing, HTTP handling, web framework compatible with node.js
- Jest - unit testing with Javascript framework allows use of mocking
- Supertest - extends unit testing by simulating HTTP request and API response
- Dotenv - manage environment variables (will be more necessary with DB connection)
- Nodemon - restarts server when files changes detected, faster development environment
- CORS - restrict requests to certain localhost ports during development and later between specified urls of frontend and backend servers during deployment of production build

List who did what within the group. TAs should be able to validate in GitHub; otherwise, team members who didn't contribute will receive a ZERO.

Group Contributions

Group Member Name	What is your contribution?	Discussion Notes
Spencer	Initialized backend server, installed and setup dependencies, created mock models for Account, Profile, Event, Notifications. Created mock instances of Profile, created Profile management "module"(controller) with unit tests to validate, routed backend and connected to	• Backend Node Server setup with: express, nodemon, dotenv, cors, Jest, and supertest • Frontend dotenv and server API hookup

	frontend. Created VolunteerMatching module(controller) with unit tests to validate, routed backend and connected to frontend(along with using ProfileController to display volunteer info). Hooked up EventManagement module(controller) to frontend on VolunteerMatching page(displays event information), Create/Edit event card, ManageEvents page (displays all upcoming events) and Volunteer History modal(event info). Hooked up VolunteerHistory(from Profile controller) into frontend modal view. Finishing touches for frontend notification api hookup.	● Profile Management module (getProfile, updateProfile), unit test and frontend hookup ● Event Management frontend hookup ● Volunteer matching module, unit test and frontend hookup ● Volunteer History frontend hookup ● Provided list of technologies and explanations of use
Sofia/Uyen	Turning CreateEditEventCard, LoginPage, Register Page into react hook form. Form validation Requirements for the pages that related to Login/Register Module. Making mocks data to test on backend site. Test within Postman with mocks data. Creating routing after successfully register and login to the corresponding pages. Api hookup login/register frontend to backend. Hookup from auth.js with frontend backend routing from .env files. Testing with unit test for Login/Registration and make sure it met the requirements.	● Form validation requirement ref ● loginRegisterController.js ● loginRegister.js ● loginRegisterController.test.js ● CreatEditEventCard.jsx module ● auth.js ● LoginPage.jsx ● Register.jsx ● Routing Login Success -> Create Profile ● Routing Register Success -> Email Verify ● .env files for both frontend and backend
Jennifer	Turned Login and Register page into react hook form. Created user page management module and notification module. Connected the notification module	● Connecting notification module to front-end ● Notification module

	to the front-end and tested to make sure it worked.	● User page management module ● Login & register page form validation ● Hooked up notifications ● Unit testing on notification module
Rainer	Created mock events, event controller, and event controller unit tests for testing getting event info and assigned attendees. Added additional fields to profile controller, some mock profiles and mock profile fields, and profile controller unit tests. Adjusted emailverification to use proper fields to handle information. Formatted assignment info page.	● Mock events ● Part of mock profiles ● Event controller ● Event controller unit tests ● Some of Profile controller (get event history) ● Some of Profile controller unit tests (get event history) ● Connected EmailVerification.jsx to the proper forms ● Part of contribution list