Flask/React CRUD App
Project Requirements

Technical Requirements - MVP

Any items marked incomplete in this section will require you to use your one redo to re-submit them. Your project must fulfill the below requirements.

	Complete?
The back-end application is built with Flask.	
The front-end application is built with React.	
PostgreSQL is used as the database management system.	
The back-end and front-end applications implement JWT token-based authentication to sign up, sign in, and sign out users.	
Authorization is implemented across the front-end and back-end. Guest users (those not signed in) should not be able to create, update, or delete data in the application or access functionality allowing those actions.	
The project has at least one data entity in addition to the User model. At least one entity must have a relationship with the User model.	
The project has full CRUD functionality on both the back-end and front-end.	
The front-end application does not hold any secret keys. Public APIs that require secret keys must be accessed from the back-end application.	
The project is deployed online so that the rest of the world can use it.	

Code Convention Requirements

More than two items marked incomplete in this section will require you to use your one redo to re-submit them. If two or fewer items are marked incomplete, the project is considered passing. Your project must fulfill the below requirements.

	Complete?
The files in the back-end and front-end applications are organized following the conventions demonstrated in lectures.	
The code in the back-end and front-end applications adheres to the coding conventions demonstrated in lectures, like using plural names for lists and arrays.	
The back-end and front-end applications do not contain dead code, commented-out sections, prints, or console logs.	
The back-end application can be used without encountering errors in the terminal. The front-end application can be used without encountering errors in the browser's console.	
The back-end application follows RESTful routing conventions for routes.	
The back-end and front-end applications are coded using proper indentation.	

UI/UX Requirements

More than two items marked incomplete in this section will require you to use your one redo to re-submit them. If two or fewer items are marked incomplete, the project is considered passing. Your project must fulfill the below requirements.

		Complete?
	The application exhibits a visual theme, like a consistent color palette and cohesive layout across pages.	
	The application is easily navigable by a first-time user. For example, navigation should be done through links instead of having to type in a URL to navigate around the app.	
	The application utilizes CSS Flexbox and/or Grid for page layout design.	
	Colors used in the app have appropriate contrast that meet the WCAG 2.0 level AA standard.	
	When editing an item, the form is pre-filled with that item's details.	
	Only the user who created a piece of data can see and interact with the UI for editing or deleting that data.	
	All images have alt text.	
	No text is placed on top of an image in a way that makes the text inaccessible.	
	All buttons are styled.	

Git and GitHub Requirements

Any items marked incomplete in this section will require you to use your one redo to re-submit them. Your interactions with Git and GitHub must fulfill the below requirements.

		Complete?
	You are shown as the only contributor to the project on GitHub.	
	The GitHub repositories used for the project must be named appropriately. For example, names like book-binder-back-end or wellness-tracker-front-end are appropriate, whereas ga-project or flask-react-project are not. The repo must be publicly accessible. Be sure to create the repos on your personal GitHub account and ensure it is public.	
	Your repos should have commits that date back to the very beginning of the project. If you start over with a new repo, do not delete the old one.	
	Commit messages should be descriptive of the work done in that commit.	

README Requirements

More than two items marked incomplete in this section will require you to use your one redo to re-submit them. If two or fewer items are marked incomplete, the project is considered passing. Your README must contain the items or sections below. Unless stated otherwise, these requirements apply only to the front-end README file.

	Complete?
Screenshot/Logo: A screenshot of your app or a logo.	
Your app's name: Include a description of your app and its functionality. Background info about the app and why you built it is a nice touch.	
Getting started: This section should include a link to your deployed app, a link to your planning materials, and a link to the back-end repository.	
Attributions: This section should include links to any external resources (such as libraries or assets) you used to develop your application that require attribution. You can exclude this section if it does not apply to your application.	
Technologies used: List the principal technologies used by your application, for example: Python, JavaScript, and any major frameworks or libraries.	
Next steps: Planned future enhancements (stretch goals).	
The back-end README file contains the name of the project and a link pointing to the front-end repository on GitHub for more details on the project.	

Presentation Requirements

Any items marked incomplete in this section will require you to use your one redo to re-submit them. Your presentation must fulfill the below requirements.

	Complete?
Present your project in front of the class on the scheduled presentation day.	
The project you present is the project you were approved by your instructor to build.	

Summary

Project meets minimum requirements and passes.	