QUICKSTART
Over 35 Years Of Technology Training

Software Engineering Bootcamp

Project 4: Nodejs Express

Introduction

The final project will help you demonstrate the skills you have learned throughout the bootcamp, and some of its components may be reused in your future projects. For instance, you will implement a signup and login procedure that is part of most web APIs and web-applications.

You will build a system according to the principles of 3-tier architecture:

1. You will create a **data layer** using a database management system, which can either be MySQL, or MongoDB.
2. You will build an **application layer** that implements the below described business logic using node.js and Express.
3. You will build a **presentation layer** using HTML, CSS, JavaScript, and the client-side framework of your choice, using some of the following: React, React Router, Redux, Angular.

Besides following the guidelines presented in this document, you have the freedom of choice in terms of technologies used and implementation details. Keep your mind and creative thinking completely free. Don't feel bound to anyones thoughts or ideas of how to complete this project.

Make sure your implementation will be rock solid, as if your were working for real employers. As you know, in the world of software-engineering, even one mistake may cost you a lot. Think of it this way. A software engineer is like a surgeon. When implementing an application, verify your work as if lives were at stake. This kind of discipline will bring you forward in your career.

Project Details

Create a solution using the 3-Tier architecture that lets users register and log in to a forum, where users may ask and answer questions. Your solution should include:

1. A database schema and some example data. For demoing purposes, choose a free online database service, or create example files that load an empty database with example data.
2. An application layer written using node.js and express.js. I recommend creating a JSON API that accesses the database fields and prepares a JSON string for the client
3. A single page application (client-side rich web application) that communicates with the application layer by sending and receiving JSON data. You may choose the corresponding frameworks and libraries in the solution.

Instead of creating an application that would store any question on any topic, invent a theme for your application. Create a relatively small hierarchy of topics, and name your application after the

organizing principle of your application. For instance, you may create a project called PetLand, and store questions on dogs, cats, and rabbits. Alternatively, you can create a project called FrontEnd frameworks, and store questions on JavaScript, Angular, and Vue.

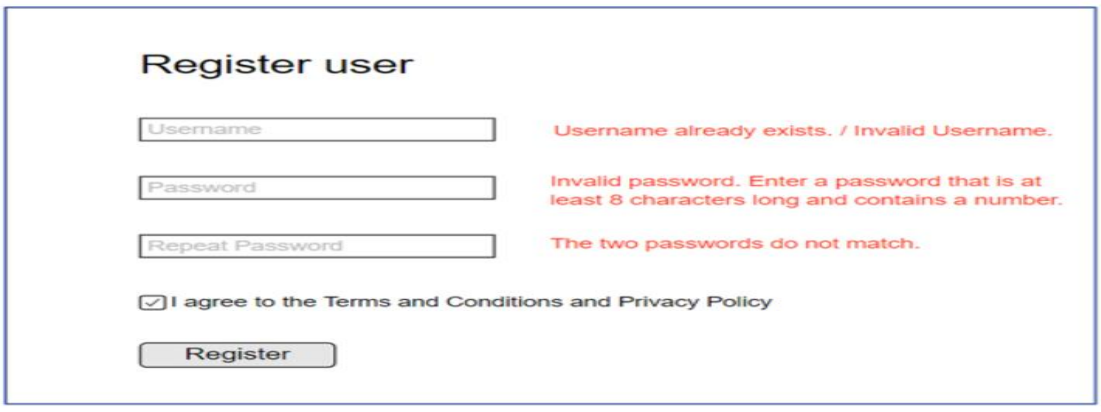
Login Screen

The login screen has two fields to enter the user name and the password.

When the username-password combination is invalid, an error message informs the user on the invalid login. If the server finds the login to be successful, the Dashboard page is shown.

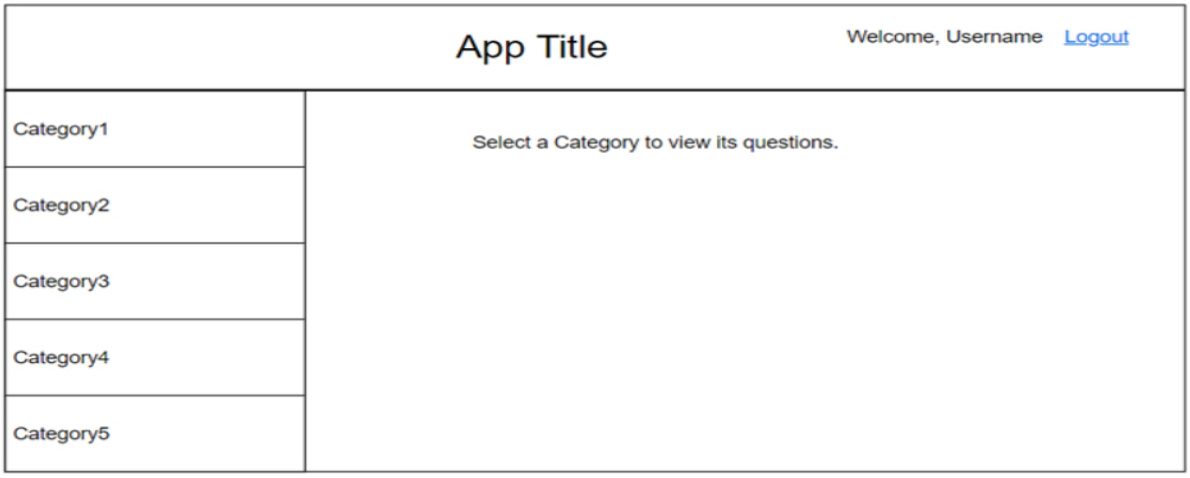
Registration Screen

The error messages appear on the right if the corresponding form field has an invalid value. The checkbox context should also be red in case its value is not checked upon pressing the Register button. Each error message should disappear once the value of the corresponding form field changes.



Dashboard

The main page contains the title of the app, displays your username, and a Logout link. The Logout link transfers you back to the Login page.
On the left, there is a menu of the existing categories. If there are more categories than what fits on screen, make sure you can scroll in this menu.
Upon selecting a category, the ”Select a Category to view its questions” area is replaced by questions in chronological order:



Deliverables

Submit your GitHub link and hosting link (include everything on a word document and upload the document under Project Submission section). All deliverables should be included on GitHub and the site should be made publicly available using a hosting service.

Project Grading

To pass the project, Instructor should take below criteria into consideration while grading this project and decide whether to Pass or Fail the student.

- **Functionality**
- **Robustness**
- **Creativity, styling, user experience**
- **Code quality.**
- **GitHub structure**
- **Documentation, Installation instructions, Comments**