

( / )



Curriculum

**Short Specializations** ^

Average: 97.3%



# Portfolio Project Idea Bank

## Portfolio Project Idea Bank

### Guide for this Project Idea Bank

- If you are in a team without Back-End engineers please use Firebase as your backend
- Please adhere to the tech stack

### Interactive Quiz Application:

#### Tech Stack:

Python (Flask/Django), HTML, CSS

#### Description:

Create a quiz application where users can answer multiple-choice questions. Implement scoring, time limits, and feedback. This project will reinforce Python programming skills and web development basics.

#### Milestones:

- Auth system/Session management
- Storage of quiz results under user accounts (This means you need to setup db storage for your application)
- Make application responsive
- Bonus: Design your code so that you're able to add new sets of quiz questions
- Bonus: Expose your quiz questions in the form of a REST API

### Algorithm Visualizer:

#### Tech Stack:

JavaScript (React), HTML, CSS

#### Description:

Build a web application that visualizes sorting algorithms like bubble sort, quicksort, etc. This project will reinforce data structure and algorithm concepts while working with React.



## Milestones:

(/)

- Feature to enable users to visualize different algorithms
- Feature to enable users to filter between types of algorithms visible (sort/search)
- Make application responsive
- Bonus: Include implementation details for the algorithm in python, C and JavaScript and allow your users to switch between them.

## Chat Application:

### Tech Stack:

JavaScript (React for frontend, Node for backend), WebSocket

### Description:

Develop a real-time chat application where users can join different rooms and exchange messages. This project will enhance understanding of WebSocket communication.

### Milestones:

- Implement authentication
- Store user chat history
- Make application responsive
- Bonus: allow users to upload profile pictures and add a bio
- Bonus: allow users to view bios of other users in the same chatrooms

## Blog Platform:

### Tech Stack:

JavaScript (React for frontend, Node for backend), MongoDB

### Description:

Build a blog platform where users can create, edit, and delete blog posts. Integrate a database (MongoDB or SQL) for storing and retrieving blog data.

### Milestones:

- Implement authentication
- Store blog posts
- Show users list of all posts they've read so far
- Allow users to create, edit and delete their blog posts
- Make application responsive
- Bonus: implement a markdown editor for users to create their blog posts on your platform
- Bonus: allow users to upload header images for their blog posts



## E-commerce Website:

### Tech Stack:

JavaScript (React for frontend, Node for backend), MongoDB

## (/) Description:

Create a simple e-commerce website with features like product listings, shopping cart, and order history. This project will involve working with databases, user authentication, and building a responsive UI with React.

- Implement authentication
- Implement storage
- Implement a working cart
- Implement product listings
- Allow users to view their purchase history
- Make application responsive
- Bonus: Integrate stripe for credit card payments (you can use the dev version)

## Continuous Integration/Continuous Deployment (CI/CD) Pipeline:

### Tech Stack:

DevOps tools (Jenkins, Docker, GitHub)

### Description:

Implement a CI/CD pipeline for a sample application. Automate the build, test, and deployment processes using Jenkins, Docker, and GitHub.

## Language Learning Flashcards:

### Tech Stack:

JavaScript (React), HTML, CSS

### Description:

Create a language learning app with flashcards for vocabulary. Include features like creating decks, flipping cards for translations, and tracking progress. This project will enhance frontend development skills.

### Milestones:

- Implement authentication
- Allow users to view cards they already went through
- Implement animation when switching from current card to the next
- Allow users to create decks of flashcards
- Allow users to share specific cards to different social media apps
- Make application responsive

## Job Board for Tech Positions:

### Tech Stack:

JavaScript (React for frontend, Node for backend), MongoDB

### Description:



Develop a job board where companies can post tech job openings, and users can search and apply for positions. Implement user authentication and a responsive design.

**Milestones:**

- Allow user authentication
- Enable users to create profile filled with information relevant to their job hunt
- Allow companies to track jobs they've posted
- Allow companies to view the number of applications they have for a job
- Allow users to track jobs they've applied for
- Make application responsive
- Bonus: Recommend jobs based on skills the user adds to their profile

**Social Media Dashboard:****Tech Stack:**

JavaScript (React for frontend, Node for backend), MongoDB

**Description:**

Build a simplified social media dashboard where users can post updates, follow others, and view a feed. Implement features like likes, comments, and user profiles.

**Milestones:**

- Allow user authentication
- Users can post and delete updates
- Users can view other users updates
- Users can follow other users
- Users can like and comment on other users' posts
- Users can create and edit their own profiles
- Users can view other users' profiles

**Online Code Editor:****Tech Stack:**

JavaScript (React for frontend, Node for backend), Docker

**Description:**

Create an online code editor that supports multiple programming languages. Users can write, run, and save code snippets (Use Docker for containerization to execute code safely).

**Milestones:**

- Implement authentication
- Users can edit their code
- Users can run their code
- Users can save and view code snippets
- Users can delete code snippets

**Health and Fitness Tracker:****Tech Stack:**

JavaScript (React for frontend, Node for backend), MongoDB

**Description:**

Develop a health and fitness tracker where users can log their exercises, set goals, and track their progress over time. Implement charts to visualize fitness data.

**Milestones:**

- Implement authentication
- Users can log exercises
- Users can set and view goals
- Users can track progress over time (implement charts for visualization)

**Event Planning Platform:****Tech Stack:**

JavaScript (React for frontend, Node for backend), MongoDB

**Description:**

Build an event planning platform where users can create, manage, and RSVP to events. Include features like location maps, date/time pickers, and notifications.

**Milestones:**

- Implement authentication
- Users can create and edit events
- Users can share links to events
- Users can RSVP to events
- Users can view the location of an event on a map app
- Users get notifications for events

**Cryptocurrency Portfolio Tracker:****Tech Stack:**

JavaScript (React for frontend, Node for backend), MongoDB

**Description:**

Create a cryptocurrency portfolio tracker that allows users to monitor their crypto investments. Integrate real-time cryptocurrency data using APIs and provide visualizations of portfolio performance.

**Milestones:**

- Implement authentication
- Integrate crypto APIs
- Implement charts to visualize portfolio performance

**Online Music Player:****Tech Stack:**

JavaScript (React for frontend, Node for backend), MongoDB

**Description:**

Build an online music player where users can create playlists, search for songs, and play music. Implement features like play, pause, skip, and volume control.

**Milestones:**

- Implement authentication
- Integrate a music API to source music

- Users can create playlists
  - Users can search for songs
  - Users can play, pause, skip, forward, rewind their music
- 

Copyright © 2024 ALX, All rights reserved.

