Josh Carter

305-896-2811
joshcarter8400@gmail.com
https://josh-carter.net/
https://github.com/JoshCarter8400
http://www.linkedin.com/in/joshcarter4813

EXPERIENCE

EMGACS, Intern - Full stack Javascript developer

Dec 2020 - Present

- Working on a team building a greenfield project for a business management and inventory system. Gaining experience with Typescript and Express with a React front end.
- Agile environment using Kanban
- Utilizing React with context and hooks on the front end to build views to integrate with the backend
- Using Typescript and Sequelize to build controllers which connect to a Postgres database
- Wrote middleware which converts all error handling to JSON
- Attend meetings with stakeholders, project managers, and developers to get insight on architectural decisions

Webefit Personal Training, Personal Trainer

April 2015 - Present

- Applying effective communication skills to conduct consultations and listening to client feedback at all times to be able to solve their individual problems allowing me to meet their needs. Creating new approaches for the business to consistently bring in new revenue

Conch Cruiser Bike Rental, Customer Service manager

April 2013 - May 2015

- Used effective communication strengths to initiate a concierge service into existing business models by building relationships with local businesses and training new employees.

SKILLS

Strong with:

Javascript (ES6)

React (context/hooks)

Node / Express

Git / Github

Heroku

Unit testing

- Jest

CSS

Bootstrap

Typescript

PWA

MongoDB / Mongoose

SQL / Sequelize

Agile Development

Worked with and learning:

- OOP
- Redux
- Inquirer
- GraphQL
- Docker
- Semantic-UI
- Vue

EDUCATION

University of Central Florida

July 2020 - Feb 2021

A 24-week intensive boot camp consisting of 20-30 hours a week of coding, weekly assignments, and 3 team projects. Learned tools and skills used in modern web tech stacks, agile methodology, and common practices seen in the workplace

American College of Sports Medicine, *Personal Training Certificate*

Oct 2014 - April 2015

PROJECTS

Quarantine Movie Nights

This was the first group project of boot camp. I was part of a 4 person team that built this project from scratch. A user can search for movies based on actor, actress, and genre and get a description of the search results allowing them to create their own watch list.

- Designed the layout for the front end of the project using Bulma which we were not taught in class.
- Learned how to use git more effectively and resolve merge conflicts.
- Used fetch to integrate the TMDb and OMDb apis to bring in movie data
- Implemented error handling functions to display errors to the user

Tech stacks utilized

- Bulma framework, javascript ES6, CSS, AJAX
- Deployed project on github pages

Demo: https://joshcarter8400.github.io/quarantine-movie-nights/

Code: https://github.com/JoshCarter8400/quarantine-movie-nights

Kill the Keg

This was the second group project of boot camp consisting of a team of 4. This app has admin and general user accounts. It allows bar owners to

advertise when their keg stock is low and they want to run a special to bring in customers. Users are able to see events and interact with other users by commenting and planning their night.

- Created Sequelize models and wrote Express routes to connect to MySql database
- Worked in agile development environment and effectively communicated with one another remotely to plan project
- Utilized bootstrap and handlebars together to create front end
- Used Environmental variables to protect sensitive information while using express sessions to store user data for persistent use
- Used photoshop to create images for front end
- Used git workflow to avoid merge conflicts
- Used Slack and zoom to work remotely as a team
- Used authentication to verify all users and to redirect to proper page if not verified

Tech stacks utilized

- MySql, Sequelize, Node, Express, Bootstrap, handlebars, Express Session, Restful API, Photoshop
- Deployed project on Heroku

Demo: https://obscure-eyrie-90813.herokuapp.com/

Code: https://github.com/JoshCarter8400/kill-the-keg

Web-Buffet

This is our 3rd and final project of the UCF Full Stack boot camp. This is an application that allows users and guests to come and view the web development service that we provide which they can then choose from and purchase. Logged in users are able to add services to their cart, leave reviews for services, and checkout via the Stripe API. Guests are able to view the team, services, and reviews on services.

- Created database models using Mongoose and used Express routes to connect to MongoDB databases.
- Used React for the front-end implementing React-Router for navigation throughout the app.
- Used Redux for state management throughout the app.
- Implemented React Semantic-UI for all styling and layouts using inline styling.
- Used GraphQL for making API calls for bringing data from the server to the client.
- Implemented Apollo Server Express for server setup and Apollo

Boost to manage data.
- Setup the checkout process with the Stripe API
- Used JSON web tokens for authentication of users identity
- We added in CSS animations for our turnstile logo on the homepage.
- Worked within git flow and used Kanban boards to prevent any merge conflicts and to create a cohesive team always working on the same page.
- Used Slack and Zoom for our team to work together remotely and efficiently.

Tech stacks utilized

- MongoDB, Mongoose, Express, Node, React(hooks), React-Router, Redux, React Semantic-UI, Apollo Server Express, Apollo Boost, GraphQL, JSON Web Tokens, Stripe API, CSS Animations
- Deployed on Heroku

Demo: https://glacial-atoll-63430.herokuapp.com/

Code: https://github.com/JoshCarter8400/web-buffet

Book Search Engine

This is a solo project I did for class that allows users to signup and search for books using the Google books API. Users can search for books by category and save them to their favorites list and they have the ability to delete them as well.

- Setup Apollo Server to use GraphQl queries and mutations to fetch and modify data
- Created an Apollo Provider so that request can communicate with the Apollo Server
- Created an authentication middleware to to work with GraphQL
- Created typeDefs to define types, queries, and mutations
- Created resolvers to generate the response for GraphQL queries and mutations
- Used React Hooks to bring in data to front end

Tech stacks utilized

- React, Express, Node, Mongoose, React Router, Apollo Server, Apollo Boost, GraphQL
- Deployed with Heroku

Demo: https://salty-mesa-30136.herokuapp.com/

Code: https://github.com/JoshCarter8400/book-search-engine