Suman Khadka(Neil)

P.316-616-7653 neeilkamal308@gmail.com LinkedIn Github Portfolio San Francisco, SF

SKILLS

JavaScript, React, Redux, HTML, CSS, Ruby, Ruby on Rails, Mongoose, MongoDB, Node.js, Express.js, SQL, SQLite3, PostgreSQL, Webpack, jQuery, Git, Heroku, AWS S3

EXPERIENCE

Full Stack software developer

Diversity, 3 months internship

- Worked on the frontend portion of a chrome extension
- Transformed mockups/prototype made by UX team into responsive HTML/CSS/REACT.JS/JSy
- Built the backend and the database using FireBase using many SDKs from FireBase like their real-time storage system and User Authentication UI to build chrome extensions.
- Created the backend and database of the chrome extension that collects and stores the text, video url , and image with option for the title and description for material in the firebase database system.

Quality Assurance Officer : (*Makalu Airlines, Nov* 2013- Dec 2016)

Structural Engineer: (*Quest Global* pvt ltd - May 2013)

PROJECTS

Virtual_shop Live Site | Github Link

This is the e-commerce site that provides the online platform for the buying and selling products. That has the information about the products, pictures related to the products and all the other details. Users can have their own store and unique cart.

- Ruby on rails was used for the backend for it's customized and built in features for the web application, Postgres sql tables were used to store data due to it's convenient way to associate the data in the rails models and also filter required data through inbuilt search query.
- Ajax was used to hit the rails controller to retrieve, store, and update the data in the database for its speed , flexibility with varied programming languages and compatibility with the JSON data format.
- Redux store is used to retain the data received from the database in the front end. The short term retention of the data and flexibility to manipulate and update the store helped to create the dynamic user interaction in the front end, which has very minimal time lag.
- The react component was used to render the information from the database through redux. Its compatibility with css and Javascript provides all the Javascript DOM events manipulations together with smart re-render feature of react provides the instantaneous user interaction with the front end view and dynamic manipulation of back end database.
- Amazon Active Storage (AWS S3) was used to store all the pictures related to the products for the quick retrieval of data and infinite server capacity. It speeds up the application and provide the flexibility to scale gracefully as our User Base grows

MyItenary Live Site |Github Link

This app allows the user to plan a trip by providing all the information about the flights, events in-between, and the housing.

- Non relational database (MongoDb) was used to store the data, since it provides the great flexibility to connect the seemingly unrelated data, and expand the schema easily
- The React full-calendar library was used to display the list of the trips on the profile of the user, clicking on the day takes the user to the list of his trips and their details. onClick event was used to store the data
- Harnessed the unidirectional state management of redux ,for the smooth rendering of the react component, and directly manage for form inputs and form submission
- Amazon Active Storage (AWS S3) was used to store all the pictures related to the products for the quick retrieval of data and infinite server capacity. It speeds up the application and provide the flexibility to scale gracefully as our User Base grows

SideMe Live Site | Github Link

It is the web version of the traditional sliding puzzle game where we need to reorganize the jumbled pic parts. It provides the option to choose out of many pictures and also the difficulty level of each game. It tracks the number of moves to make the game interesting and competitive. We can also the restart the game my reshuffling the picture portions if we hit the dead-end

- JavaScript Class was used to interact with the display, instance variable were used to represent the grid of the game in the frontend and constructor functions were used to store the working logics for the game and operation
- HTML features h, p, div, br were used to create the structure of the display component and CSS features were used to decorate the structure. CSS feature background-image was used to display the slides of the pic
- JavaScript DOM feature add class and remove class was used in combination to the onClick event handler to toggle the class between the HTML div (box) to make image slide on click
- Add element and Remove element features was used in combination to the onClick event handler to dynamically change the display
- JavaScript DOM Alert feature was used to display the important message to the user.
- Firebase was used for storing images, that made the game more dynamic since users can choose images of preference.
- All the images Url addresses were stored as the class constant so that we can have access to those in every instance of the class.

EDUCATION

App Academy - Immersive software development course with focus on full stack web development (2019)

Wichita State University - One semester - MS - aerospace engineering (major computational Fluid Dynamics)

VTU, Karnataka, Inda- *BE - Aeronautical Engineering -2009-2013*