ADNAN NIAZ

E: sanixstudio@gmail.com
P: 415-802-8751
https://bit.ly/3wB9ncG

I am a full-stack engineer. I solve problems and bring designs to life. If you are curious, please take a look at my work:

GitHub: Github.com/sanixstudio/
LinkedIn: Linkedin.com/in/sanixstudio
CodeSandBox: codesandbox.io/u/sanixstudio
Replit: Replit.com/@sanixstudio

TECHNICAL SKILLS

Front-End: HTML5, CSS3, SASS, JavaScript, jQuery, React, Bootstrap
Back-End: Node, Express
Databases: MySQL, MongoDB, Firebase
Design: Balsamiq, Figma, Photoshop
Other: Github, REST, HTTP standards, Agile and Waterfall methodologies

WORK EXPERIENCE

Software Engineer (Contract) - Task-Buddy (San Francisco, January 2021 - Present) (demo | code)
- Created a fully responsive application for Task-Buddy using HTML5, CSS3, and pure JavaScript
- Refactored existing code to make components responsive and add new design layout
- Collaborated with other engineers, pair programmed, and gave existing design new life based on styled guide provided by a designer

Front-End Engineer - Code for America (San Francisco, July 2020 - November 2020)
- Contributed to various projects for code for America brigade in San Francisco, Bay Area
- Implemented UI/UX features using HTML5, CSS3, and JavaScript
- Additionally applied modern EcmaScript-6 standards such as destructuring, rest/spread operator, writing clean and readable code that is scalable

PROJECTS

InfoMovia (demo | code)
- Using **HTML5**, **CSS3** and vanilla **JavaScript**, created a front-end application called Infomovia that allows you to retrieve information for movies/tv-shows using a third party **API** called TMDB.
- Infomovia is fully **responsive** and **mobile friendly**, it leverages **flexbox** and **media queries** to respond to the size of the window
- Applied custom styling and design using **Figma**
- Validating user string to handle errors and exceptions, preventing app crash

TODO APP (demo | back-end | front-end)
- Using **Node/Express, React**, and CSS framework **Bootstrap**, created a full-stack Todo application
- The Todo application features full **CRUD** (create, read, update, and delete) functionality
- When users create, read, update or delete, the front-end (React) makes an API call to the Node/Express back-end which then updates the **Mongo database** and as a result sends valid **HTTP** response back to the front-end
- Lastly, Todo is deployed on Heroku

FireGallery (demo | github)
- Using React, **Firebase**, and **Firestore**, created a front-end image gallery application called FireGallery
- FireGallery allow users to upload images that are seen by the community members that are active on FireGallery
- One of the best features of the FireGallery codebase is that it leverages **React hooks**, **code modularity** and **clean code** practice. In addition, in order to make the images interactive and pop out, framer motion is used for **smooth transition**
- Used Framer Motion for smooth transition effect
- FireGallery is deployed using Netlify

EDUCATION
Full-Stack Developer, **UC-Berkeley Extension** (San Francisco Bay Area)
Front-End Developer (**Scrimba**)