PARIK KAPADIA

parikkapadia21@gmail.com | 217-974-5908 | Champaign, Illinois | US Citizen | Linkedin | Github | Portfolio

SUMMARY

Versatile **software engineer** with more than 5 years of experience across domains like scientific programming and more recently full stack app development. Thrives in fast-paced, technologically progressive environments.

SKILLS

Languages: Python, JavaScript
Backend: Node.js, Express.js, Django, Flask
Frontend: React.js, React Native, Angular, HTML, CSS
Databases: SQL (MySQL, PostgreSQL), NoSQL (MongoDB)
Cloud: AWS, Azure, Google Cloud (GCP), Docker, Kubernetes
Machine Learning: TensorFlow, Keras, PyTorch

EDUCATION

University of Illinois at Urbana-Champaign Champaign, IL
Master of Science (MS) in Statistics 01/2019-08/2020
Bachelor of Science (BS) in Electrical Engineering 01/2016-12/2018

PROFESSIONAL DEVELOPMENT

CareerFoundry Remote
Full Stack Web Development Program 07/2023-11/2023

INDUSTRY EXPERIENCE

Wolfram Research Champaign, IL
Software Developer 01/2017-12/2020

- Served as a high-level programmer in the Algorithms Research and Development department for the Mathematica product.
- Contributed to the development of functions fundamental to numerical analysis, optimization, signal processing, differential equations, linear algebra, vector calculus, probability, statistics and machine learning.
- Verified functionality of 30+ built-in advanced mathematical and statistical functions in Mathematica product by coding test suites of 20K+ black box test cases based on problems researched from 100+ textbooks and translated into high-level code.
- Implemented packages for new and existing functions including Fourier and Laplace transforms by coding 1K+ test cases to ensure production of accurate output of complex inputs and modifying code to increase their versatility.
- Helped the software development team improve the reliability of these functions by 5 to 10 percent during the release cycles of Mathematica versions 11 and 12.
- Collaborated with the software quality assurance team to run and evaluate 1K+ regression test cases to ensure correct output of existing functionalities.
- Identified and wrote high-level prototype code for 10+ highly detailed mathematical functions to be potentially developed further and added to later releases of the product.

SOFTWARE ENGINEERING PROJECTS

Movie Data App | 08/2023-10/2023

- **Backend:** Developed backend API to store movie and user data using Node.js runtime environment and Express.js framework. Created a REST API for sending requests for movie and user data to the server in JSON data format. Set up database in NoSQL (MongoDB) for movie and user datasets and integrated it with REST API using Mongoose. Tested REST API endpoints using Postman. Enhanced data security and validation by adding username and password authentication features using Passport.js. Deployed app backend to Render web hosting service. Link
- **Frontend (React):** Developed client-side frontend for app using React.js and integrated it with the backend API for fetching data. Incorporated functionalities for users to sign up, log in and out, search for, filter and view details of movies, add them to favorites and delete them, display user details, update and delete account and routing to navigate between viewing pages. Applied Bootstrap to replenish user interface (UI) styling. Deployed frontend to Netlify web hosting service. Link
- **Frontend (Angular):** Built a client-side for the API using the Typescript framework Angular. Implemented functionalities and routing for the app including a welcome page, forms for the user to sign up, login, view and update their profile information, view a list of movies and their details including genre, director and description. Pending functionalities to be developed include adding and deleting movies to a list of favorites as well as images to be assigned to individual movies. Auto-generated documentation for both the API and frontend code using JSDoc and TypeDoc for future developers to refer to. Link

Messenger Chat App | 10/2023-10/2023 | Link

- Built mobile app in React Native for a messenger that allows users to chat and send messages to each other. Set up database in Google Firestore (NoSQL) and Firebase for storing message data containing text and images in real time and for app authentication. Incorporated capabilities for using app offline and storing cached data, capturing and allowing a user to send image, video and location data. Simulated and tested app in Expo (Go) and Android Studio emulator.

Event Calendar App | 09/2023-10/2023 | Link

- Developed client-side frontend for app in React.js to load data on events being held in various cities around the world from an existing API into a calendar. Constructed user stories in Gherkin and Cucumber to describe functionalities of app features and define its requirements. Wrote and tested serverless functions with AWS Lambda to efficiently handle API requests and administered access tokens with Google OAuth to allow users to access the app. Wrote unit tests, integration tests, user acceptance tests and end-to-end tests using Jest and Puppeteer to test the functionalities of the app components individually and as a whole including event details and their lists, search bar for event filtering, alerts for invalid user input and API data connection and to validate the user stories and achieved 100 percent pass rates. Implemented responsive chart displays for visualizing aggregations of event data.

Recipe App | 11/2023-11/2023 | Link

- Developed a full-stack (backend and frontend) for an interactive app to upload and view recipes in Django and implemented using MVC architecture. Implemented functionalities for a user to add and delete recipes and specify their characteristics as well as to view visualizations of statistics regarding the data of the recipes and search for them. Set up data model schema for inputs, user authentication, views and templates for frontend and wrote test cases for input validation. Deployed to Render hosting service. Coded in Python.

MACHINE LEARNING PROJECTS

NLP Modeling for Question Answering Dataset | 11/2023-11/2023 | Link

- Built a natural language processing (NLP) model for answering questions using the deep learning libraries TensorFlow and Keras on CPU by applying the large language model (LLM) and transformer BERT and a tokenizer on 5 questions from a text dataset containing 10K questions from Hugging Face API. Tuned hyperparameters of 3 epochs, batch size of 1 and applied Adam optimizer to accomplish task. Trained and validated model and achieved high accuracy on validation and train sets and a low loss function metric for the hyperparameters picked. Pending improvements to model include increasing dataset size and batch size and decreasing learning rate once computing resource and GPU availability is increased. Coded in Python.

Large Scale Machine Learning Modeling for GPU Performance | 05/2023-05/2023 | Link

- Processed, transformed, manipulated and analyzed large dataset of 240K rows in Spark (PySpark). Implemented, trained and tested predictive machine learning model using gradient boosting regression and set up pipeline components to predict the performance of a GPU kernel using 14 variables. Tuned hyperparameters to optimize model performance and achieved low error metric for model. Coded in Python.

Predictive Modeling on Large Dataset for Diabetes | 01/2022-01/2022 | Link

- Explored, analyzed, wrangled and cleaned large dataset containing 100K+ rows and 50+ columns of data using Pandas library in Python. Applied machine learning techniques including KNN, logistic regression, decision tree, random forest using Scikit-learn library in Python on dataset to develop predictive models and set up a pipeline for both processing and modeling. Trained and tested models and received accuracy and low error rates less than half for all.

Gradient Boosting Modeling on Dry Bean Dataset | Link

- Built machine learning models in XGBoost and LightGBM to predict the class of a dry bean based on its numeric features from a dataset with 10K+ rows. Achieved 90+ percent accuracy for models. Coded in Python.