

Vama Shah

Rochester, NY • vamashah21@gmail.com • (585) 410-7319 • www.linkedin.com/in/vamashah21

TECHNICAL SKILLS

Languages: Python, Java, SQL, C/C++, JavaScript, Matlab, Angular

Database: MySQL, Oracle DB, MongoDB

Analytical: Machine Learning, Artificial Intelligence, NLP, Data Structures and Algorithms

Tools: ThreeJS, BitBucket, Jira, GitHub, Jupyter Notebook

Libraries: PyTorch, Numpy, SciPy, Tensorflow, Pandas, Scikit-learn

EDUCATION

Rochester Institute of Technology, Rochester, NY

Aug 2020 - May 2023

Master of Science, *Computer Science* GPA: 3.5/4.0

K. J. Somaiya College of Engineering, University of Mumbai, India

Aug 2015 - May 2019

Bachelor of Technology, *Electronics and Telecommunication Engineering* GPA: 7.34/10

PROFESSIONAL WORK EXPERIENCE

Software Development Engineer Intern

Jan 2022- Present

MINDEX Technologies Inc., Rochester, NY

- Responsible for developing and testing software with 90% code coverage to process company tax information from 23 states and reduce tax burden for the companies by 66%.
- Proposed requirement for priority status of incoming files and implemented it to increase processing speed by 80%
- Developed new modules and enhanced existing ones to design flow-based solutions between UI and Mongo Database for client **Paychex** based on **Agile** methodology
- Technologies- **Java 6 and 8, Spring Framework, Gradle, Maven, Jenkins, AngularJS, TypeScript**

Assistant Systems Engineer

Nov 2019 – Dec 2020

TATA CONSULTANCY SERVICES (TCS), Mumbai, India

- Worked on development of in-house Data Science Platform for client
- In-charge of onboarding 80+ users and developing Python scripts to automate user-access to the platform
- Focused on automating specific project requirements to use the platform which included automating the creation of virtual environments with the required set of packages which reduced set-up time by 95%
- Technologies- **Python, Django, SQL**

PROJECTS

Topic Modelling for Olympic and Fashion News <https://github.com/vama-rit/Topic-Modelling-for-Olympic-and-Fashion-News>

Sept 2021

- Identified distinct topics occurring in large sets of documents based on the frequency of words present in them
- Used LDA model to perform modelling and measured the performance using coherence and perplexity
- Identified topics after data cleaning and data pre-processing with 89% accuracy

Binary Classifier using PyTorch

Sept 2021

- Created a binary classifier using PyTorch to implement a single logistic regressor
- Obtained decision boundary using the classifier to visualize class regions with classification accuracy of 98%

Optimal Path Generation for Orienteering

Feb 2021

- Found the shortest path between two coordinates on a 3D map by performing the A* search algorithm which is 40% faster than any naïve search algorithm
- Considered optimal path for orienteering considering different types of terrain and elevations and used BFS to detect water body edges which improved performance by 20%

ASL Gesture Recognition and Conversion from text to speech

Jan 2019

- Built and trained a Convolutional Neural Network to recognize hand gestures with 94% accuracy
- Employed image processing techniques to filter incoming hand images and make them easier to train on/get recognized by the neural network
- Used Tensorflow library for implementation of the deep learning model and understood its various components

HACKATHONS

Neighborhood Evacuation WebApp- <https://devpost.com/software/neighborhood-evacuation>

Spring 2022

BrickHacks 2022

- Developed a web app that allows users to post location determining availability of shelter, supplies or transport
- Technologies used- **Python, Django web framework, Cloud services and APIs**

EmployeeFit Productivity Tool- <https://devpost.com/software/employeefit>

Spring 2022

WicHacks 2022

- Implemented a decision tree machine learning algorithm to determine work preference of work from home or office and obtained a 76% accuracy for determination of work preference
- Technologies used- **Python, Mental Health of Employees dataset from Kaggle**