ZAHIN M. ZAMAN

✉ zm2zaman@uwaterloo.ca

📞 (519) 721-2837

in linkedin.com/in/zahin-zaman

👤 zahin-zaman.vercel.app

 github.com/alvii147

 devpost.com/alvii147

PROJECTS

Altruist

- Mobile and web platform to promote altruism and the assistance of senior citizens during a difficult time
- Assembled **Django** server with REST API using **Django REST Framework** including user login and registration
- Designed and constructed elegant, cross-platform mobile application using **React Native** and **Expo**

cram.ai

WinterHacklympics – **Best Use of Google Cloud**

- Web platform that uses natural language processing to analyze lecture videos and generate learning flashcards
- Incorporated **NLTK** and **Google Cloud Language** to summarize lecture videos
- Generated flashcards with questions and answers in a user-friendly frontend served by a **Flask** backend hosted on **Heroku**

k kard

Hackwestern – **Perfect Pitch** & **FuturFund: Women in Financial Literacy Award**

- A mobile app that provides a fun, yet educational way for children and parents to manage their finances
- Developed & deployed fin-tech mobile application with stack and tab navigation using **React Native** and **Expo**
- Assembled a document-based **Firebase** backend to store user credentials

EduSource

HackTheU – **Best Use of Google Cloud**
Hackrithmitic – **Best Use of Google Cloud**

- Web application for enriched remote education, crowdsourcing course materials and scientific equation recognition
- Employed **Google Cloud Vision** to implement scientific equation recognition from handwriting
- Built **Flask** app with **HTML, CSS,** and **Bootstrap,** and managed database **ORMs** in **Flask SQLAlchemy** and **SQLite3**

TECHNICAL SKILLS

Languages: Python, C++, Go, JavaScript, HTML, CSS, Bash
Frameworks: Django, React.js, MySQL, PostgreSQL, Flask, Ruby on Rails, NumPy, pandas, TensorFlow, Matplotlib
Tools/Services: Docker, Kubernetes, AWS, Google Cloud

EDUCATION

University of Waterloo

B.A.Sc. in Electrical Engineering, 4th Year | Sept 2018 – May 2023

- **Term Dean's Honour List,** for outstanding academic performance
- **Artificial Intelligence Degree Specialization**

EXPERIENCE

Full-Stack Developer

Prodigy Education | Jan 2022 – April 2022

- Maintained **OAuth 2.0** & **OpenID Connect** identity service built on **Ruby on Rails**
- Leveraged multi-staged **Docker** builds to optimize production server container hosted on **Amazon ECS**, reducing image size by **58.4%**
- Designed & documented **Apache Kafka** calls to stream backend server events for data tracking and verified data payloads with unit tests utilizing **RSpec** & **Minitest** on Rails

Full-Stack Developer

Nokia | May 2021 – August 2021

- Developed & managed authentication security, community articles page, and voucher redemption system for Nokia Network Developer Portal on **Django** server with an **Azure MySQL** database
- Secured backend using **Django REST framework** API permissions, cross site scripting protection, and honeypot setups
- Composed unit tests for Django forms, models, and API endpoints, and configured **Docker** image for **Gitlab CI** automation testing

Open-Source Software Developer

codePrentice | Sept 2020 – Present

- Expanded Python multiparty-computation library, **MP-SPDZ,** to support **CNNs** including SqueezeNet, ResNet, and DenseNet
- Composed comprehensive tutorial based on Matrix Profile research paper for Python time series analysis library, **STUMPY**
- Optimized cache utilization in STUMPY's multi-threaded Matrix Profile computation by implementing tiling scheme algorithm using **NumPy** arrays and **Numba** just-in-time compilation

Software Developer

Wind River Systems | Sept 2020 – Dec 2020

- Devised thread-safe functions in **C** and inline **Assembly** to fix multi-threading and memory-based defects for VxWorks RTOS
- Developed interactive application in **PyQt5** that assists in writing git commit messages and verifies status of Jira issues and code reviews

Display Verification Engineer

Qualcomm | Jan 2020 – May 2020

- Attained **20%** increase in functional coverages by engineering **SystemVerilog** assertions to verify processor design
- Automated formal verification using **Perl** scripting to extract design hierarchy and formulate assertions