Ramneek Riar

289-242-4692 | ramneek.kaur@ryerson.ca | linkedin.com/in/ramneekriar | github.com/rkriar

Education

Toronto Metropolitan University (formerly Ryerson) Expected Graduation May 2024
Bachelor of Science in Computer Science (Co-op) *Cumulative GPA: 3.80/4.33*

Coursework: Data Structures & Algorithms, Object Oriented Programming, Discrete Mathematics, Introduction to UNIX, C, and C++, Linear Algebra, Probability and Statistics, Operating Systems

Technical Skills

Languages: Python, JavaScript, Java, C, HTML/CSS

Technologies: Git, BitBucket, Confluence, scikit-learn, Pandas, NumPy, UNIX, Selenium, MUI

Experience

Quality Assurance Analyst Intern | BMO Financial Group *May 2021 – December 2021*
Design Experience Team *Toronto (Remote)*

- Developed a suite of test cases from scratch to perform automated visual regression testing for over **70+ web components** using **Python Selenium** and Applitools in agile environment
- Refactored the team code base by **100%** as a result of updating to latest version of **Storybook** to improve efficiency of components with **InVision**
- Analyzed components to unify **100+ pages** of documentation and code base between subsystems on InVision
- Maintained and achieved accessibility guidelines **(WCAG 2.1**) whilst optimizing web components collaborating with design and development teams

Projects

Notion Library Alexa Skill | *JavaScript, Alexa Skills Kit, Notion API, Google Books API*

- Designed and implemented an Amazon Alexa skill that leverages Notion API and Google Books API to update my personal library database in real time using AWS Lambda
- Programmed intent handlers to parse slot values and generate async requests to add, retrieve or update properties such as author, status, genre, rating, format and number of pages for a given book
- Created custom slot types for sample utterances to train and test model utilizing Alexa Developer Console

Spacestagram | *React, NASA API, axios, MUI*

- Developed and deployed an Instagram clone web-app with NASA API to display posts with infinite scrolling
- Used async functions to fetch and handle data as Post objects when user reaches end of page
- Designed loading state and error-handling UI dependent on API response and utilized MUI Design Library components such as tooltip, popover, buttons and icons
- Implemented features allowing user to like/unlike posts to view their liked posts from local storage and click to copy image url

Command Line Hangman | *Python, Wordnik API*

- Recreated the game *Hangman* using Object Oriented Programming and Wordnik API to fetch random word
- Provided user with prompts to guess a letter, kept track of guesses and updated display with visual progress throughout game