Harsh Moradiya

Windsor, ON | (226) 961-5961 | moradi21@uwindsor.ca | linkedin.com/in/harshmoradiyace/ | github.com/HaRsH8747

TECHNICAL SKILLS

- **Programming Languages:** Java, Kotlin, C, Python, Javascript, PHP, C++
- **Markup Languages:** HTML5, CSS3, XML
- **Android Development:** Android SDK, Retrofit, Room, Kotlin Coroutines, RxJava, Dagger/Hilt
- **Web Development Frameworks/Libraries**: ReactJS, Django REST Framework
- **UI/UX Design:** XML Layouts, Material Design 3, Figma, Adobe XD, Tailwind CSS
- **Database:** SQLite, MYSQL, Firebase, MongoDB
- **Testing:** JUnit, Espresso, Selenium (including web scraping)
- **Tools/Technologies:** Postman, GitHub, GitLab, Bitbucket, Jira, Jenkins, Slack, Trello, Word, Excel, PowerPoint
- **Soft Skills:** Communication, Teamwork, Problem-solving, Adaptability, Creativity, Critical thinking, Report Writing

EDUCATION

Master of Applied Computing Sep 2023 – Dec 2024
University of Windsor, Windsor, ON

Bachelor of Engineering in Computer Engineering Jun 2018 - May 2022
Shree Swami Atmanand Saraswati Institute of Technology, Gujarat, India

PROFESSIONAL EXPERIENCE

ANDROID DEVELOPER Jun 2022 - Jul 2023
Twinnet Technologies, India
Technologies: Java, Kotlin, Android SDK, Retrofit, Room, MVVM, JUnit, Espresso, Slack, Jira, Trello

- Engaged in regular **Scrum** meetings, contributing to project synchronization, task prioritization, and iterative development
- Applied **MVVM** architecture to keep code clean and easy to understand
- Employed **Dagger** to efficiently manage app components, facilitating testing and optimization
- Architected and deployed **Kotlin coroutines** to streamline background task execution, enhancing app performance by **reducing lag** time by **50%** and improving overall user experience
- Conducted **code reviews, unit tests**, and **debugging** sessions to maintain code quality and stability
- Managed **Jenkins** instance for continuous integration and deployment of the company's **Java-based product** advertisement library, **automating build, test,** and deployment processes
- Conducted **pentesting** to assess API vulnerabilities utilizing **Java decompiler, Smali injection**, and reverse engineering tools, reducing the risk of unauthorized access and **mitigated** unknown API hit requests **by 20%**
- Spearheaded the research and adoption of cutting-edge product analytics software, leading to a **30% growth** in **user engagement metrics** and a **20% uptick** in customer satisfaction scores

ANDROID DEVELOPER INTERN Jan 2022 - Mar 2022
Techeshta, India
Technologies: Java, Kotlin, Android SDK, Retrofit, Room

- Gained hands-on experience in Android application development and expanded proficiency in Java and Kotlin programming languages, **Android SDK, Retrofit, Room**, and **MVVM** architecture
- Published applications on the **Google Play Store** with **6k+ total** downloads, showcasing practical application of acquired skills in real-world projects

ACADEMIC PROJECTS

Farm Tech Backend
University of Windsor, Windsor, ON Jan 2024 – Apr 2024
Technologies: Python, PostgreSQL, Django Rest Framework, scikit-learn
github.com/HaRsH8747/farm_tech_backend

- Engineered a scalable backend system using **Python** and **Django**, implementing **RESTful API**
- Implemented a crop recommendation system using scikit-learn's **Random Forest algorithm** and Python for personalized crop suggestions based on environmental and regional data with **90%** accuracy

Farm Tech
University of Windsor, Windsor, ON Jan 2024 – Apr 2024
Technologies: ReactJS, Tailwind CSS, HTML, CSS, Javascript, Axios
github.com/HaRsH8747/farm-tech-frontend

- A combined collaborative platform for farmers offers solutions such as land partnership, post-harvest storage, and personalized crop recommendations
- Developed a user-friendly **React** frontend with **Tailwind CSS**, enabling sustainable agriculture through land optimization, waste reduction, and informed crop selection
- Integrated **Axios** for consuming authenticated **RESTful APIs**, handling JSON data, and implementing **JWT authentication** in the application

Client-Server Architecture
University of Windsor, Windsor, ON Jan 2024 – Apr 2024
Technologies: C, Socket Programming, Linux Environment, Process Control
github.com/HaRsH8747/Client-Server-Architecture

- Developed client-server application using **UNIX/Linux** and **C socket** library
- Used **fork** to handle multiple client requests
- Managed **server balancing** by redirecting traffic to mirror servers

Car Rental Analysis Sep 2023 - Dec 2023
University of Windsor, Windsor, ON
Technologies: Java, Selenium, DSA, Jsoup, **Regex**
github.com/harshmm21/Car-Rental-Scraping-Analysis

- Utilized Java's core **data structures and algorithms**, including **Trie, B-Tree, Priority Queue, HashMap**, and **MaxHeap** for efficient spell checking, inverted indexing, page ranking, and frequency analysis
- Employed **Jsoup** library for robust HTML parsing and data extraction, mapping relevant information to Java objects for further analysis and storage in **JSON** format
- Leveraged **regular expressions** (regex) for data validation, pattern matching, and text processing tasks, ensuring data integrity and facilitating information retrieval

Store Locator Dec 2020 - Apr 2021
Shree Swami Atmanand Saraswati Institute of Technology, India
Technologies: Kotlin, Android SDK, Google Maps API, Firebase
github.com/HaRsH8747/Store-Locator

- Developed a **digital marketing** solution for small shops to **boost visibility** and customer engagement
- Leveraged Firebase services like **Firestore Database, Storage, Authentication** for real-time data management and secure user authentication
- Integrated **QR scanning** feature, enabling customers to redeem discounts on products by scanning QR codes
- Deployed an algorithm to prioritize products based on customer purchases, **elevating popular items** to the top of search results

PERSONAL PROJECTS

Wallstick - HD Wallpaper Platform Aug 2021
Technologies: Kotlin, Retrofit, Room Database
github.com/HaRsH8747/Wallstick

- Used **Kotlin coroutines** to handle background tasks and provide a smooth user experience.
- Implemented **pagination**, using algorithm specifically designed for handling request for **multiple APIs** at once and as a result **reducing bandwidth consumption** and total API hits by almost **40%**
- Utilized local database storage to minimize API hits, improving app performance and responsiveness