ABHINAV GUPTA

a363gupt@uwaterloo.ca ⋄ +1(548) 333-7626

abhinavgupta.info ⋄ github.com/AbhinavGupta2002 ⋄ linkedin.com/in/abhinavgupta2002

EDUCATION

Candidate for Bachelor of Honours Computer Science, University of Waterloo — Sep 2020 - Apr 2025
3 Academic Term Distinctions
Relevant Courses – User Interfaces, Data Structures and Management, Elementary Algorithm Design, Combinatorics

LANGUAGES

TypeScript, JavaScript, C++, C#, C, Kotlin, Java, SQL, Python, Swift, HTML/CSS, LaTeX

DEVELOPER TOOLS

React, NodeJS, .NET, MongoDB, PostgreSQL, OracleDB, Elastic Search, AWS, GCP, DBeaver, Studio3T, Express, Flask, Heroku, TailwindCSS, Redux, MeteorJS, Robot Framework, Swagger, Appium, Cucumber, Jest, Mocha, Jenkins, Postman, Git

WORK EXPERIENCE

Cloud Platform Developer — May 2023 – Aug 2023
Deloitte — *Toronto, ON*

- Engineered a decoupled system to search for 2,000+ cloud bundles stored in **MongoDB** using **.NET** and **C#**, incorporating advanced partial string matching - included developing a dynamic Mongo query builder for streamlined data retrieval.
- Devised a JSON-driven service to automate identifying and scheduling the execution of tasks for cloud account provisioning.

Software Engineer — Jan 2023 – Apr 2023
TradeRev — *Toronto, ON*

- Implemented REST APIs in **AWS Serverless** with **PostgreSQL** and **Elastic Search** databases, which involved building lambda functions, handlers, controllers, query builders, post-processors, and adding on to **Swagger** files in **TypeScript**.
- Built a **NodeJS** script to run and manage multiple threads in parallel to query 6 million rows of data from **OracleDB** and generate 500,000 rows of data per file. Improved performance by **160%** when compared to single-threaded querying.
- Developed object-oriented backend services that integrate **Angular** concepts, like injection, which call external APIs from organizations like Okta and Adesa. This involved writing **Cucumber** integration and **Jest** and **Mocha** unit test cases.

Frontend Developer — May 2022 – Aug 2022
TutorOcean — *Ottawa, ON*

- Utilized **ReactJS**, **MeteorJS**, and **CSS** to develop web components to implement new features on the B2C EdTech platform, such as a tutor availability blocking system, and fix over **40%** of issues reported across all B2B SaaS products.
- Diagnosed and documented complex issues by finding patterns in data retrieved from **MongoDB** using **Studio3T**, such as students being able to leave an online session when mandatory input fields in the check-out form are left empty.

Co-Founder and CTO — May 2022 – Present
Serv2U 🔗 — *Waterloo, ON*

- Managed and technically mentored a team of 5 software developers; worked closely with the UI team on a regular basis.
- Improved the platform's performance by over **50%** by rebuilding frontend web components with **ReactJS** and **CSS**, such as using **Redux** to avoid prop drilling and creating a component library to increase reusability.
- Built a full-stack authentication system for signing in users on the platform. Integrated **JWT** authorization with APIs as middleware using **Flask** with **Python**, where refresh tokens are being updated automatically through **cron** jobs.

Test Automation Developer — Sep 2021 – Dec 2021
TutorOcean — *Ottawa, ON*

- Developed over 50 test cases for the B2C website using **Robot Framework**, **Selenium WebDriver**, and **Python**.
- Established the Mobile Test Automation division, where **Appium** is used with **Robot Framework** to build test cases.

PROJECTS

TripTailor (UofTHacks Project) 🎮

- Engineered and built a multi-table **PostgreSQL** database to store users, destinations, and their relationships, managed by **ElephantSQL**, for an AI platform that enables users to make travel plans for any country in the world with ease.
- Fetched and displayed images to the **ReactJS** frontend from the **Unsplash API**, whose queries consist of countries, cities, and locations being generated with prompt engineering through NLP by the **Cohere API**.
- Developed backend REST APIs in **Python** with **Flask** and frontend API calls in **JavaScript**; hosted on **Render** platform.