SANAT BHALLA

s24bhall@uwaterloo.ca
+1 (647) 997-6244

LinkedIn - SanatBhalla
GitHub - sanat77
sanatbhalla.me

EXPERIENCE

Associate, Cloud Engineering *Toronto, ON*
PwC Canada *Sept - Present*

- Developing a full stack web application using React.js on the frontend, Nest.js and Node.js on the backend with a MongoDB NoSQL database to perform ORM

Co-Founder & Developer *Waterloo, ON*
Crypto.ai *June 2022 - Present*

- Developing an in-progress web newsletter application based on market sentiment analysis using Natural Language Processing on cryptocurrencies
- Used React.js, Typescript and Javascript to develop the frontend
- Used Node.js, Nest.js and Express to write APIs and interact with a MongoDB NoSQL database on the backend
- Built Authentication and Authorization modules using JWT and Passport API
- Developed and Deployed ELT data and ML pipelines written in python to migrate information between different modules and apply business logic to serve the client

Embedded Software Developer *Waterloo, ON (Remote)*
Ford Motor Company *Jan - April 2022*

- Developed the UI framework for the 10", 12" and 15" infotainment systems of Ford Mustang Mach E and F-150 using React.js, JavaScript and TypeScript
- Implemented the Observable pattern in React.js to improve the code scalability of Towing and Trailer Blind Spot feature on the latest launch of Ford F-150
- Worked with micro-hooks in React.js to separate the components based on business logic and leverage the code coverage during unit testing
- Developed C/C++ modules to render React Native components on the Unreal Engine using the MQ and TCP client brokers to decrease the screen startup time by 55% compared to the web
- Developed a customized animation library (Bezier Easing) using C/C++ and used it to create loading-dots in React Native with a broader flexibility than CSS
- Made use of a customized Flatbuffer repository in C/C++ to tailor the styling attributes and completely replace CSS and enhance efficiency
-Developed unit tests for JavaScript and C/C++ modules using Jest and Gtest respectively

Cloud Infrastructure Developer *Ottawa, ON (Remote)*
Innovapost Inc. *May - Aug 2021*

- Worked on a continuous ongoing project to shift all the Canada Post data from Legacy Oracle Servers to the Cloud Databases in an Agile and Scrum format
- Developed Initial and Delta Load ELT Pipelines using Python that performed various transformations to move data from Oracle to Azure SQL Server
- Used Azure Data Factory as an Orchestrator to accommodate the pipeline and made Spark API calls to trigger Databrick notebooks
- Used Azure DevOps to automate the CI/CD pipeline for the latest builds and releases
- Developed and Documented unit tests to check the working of different parts of the Data Pipeline at the functional level

Coding Instructor *Mississauga, ON (Remote)*
CoderCookies *Sept 2021 - Jan 2022*

- Introduced basic to advanced level concepts (including OOP, Data Structures and memory management) of programming in C/C++, Python, JavaScript to intermediate students

EDUCATION

University of Waterloo

Waterloo, Canada | **2020 - 25 (expected)**

Candidate for Bachelor of Computer Science (*Specialization in SWE*) - 3.7/4.0 GPA

Member of UW Data Science Club

Member of UW Comp Science Club

DPS Kalinga

Cuttack, India | **2006 - 20**

AISSE (class 10) - 92.4%

AISSCE (class 12) - 96.6%

SKILLS

Languages

Python, C/C++, JavaScript, TypeScript, SQL, R, HTML/CSS, DrRacket

Data Science / Engineering

Scikit-learn, MySQL, Apache Spark, Databricks, MongoDB

Software/Web Development

React.js, React Native, Angular, Bootstrap, Node.js, Nest.js, UnrealEngine, Flatbuffers

Others

Linux, Git, Cloud Services (Azure, AWS, GCP)

PROJECTS

Baby Compiler | C/C++

(Private Repository, message to see)

- Built a C/C++ compiler for an interpreted language with C++ styled syntax and provided user flexibility to dynamically store and free memory using the heap
- Implemented parts of the compiler including Scanner, Parser, Context Sensitive Analysis and Code Generation
- Used the SLR(1) parser for efficient Bottom-Up Parsing
- Used MIPS as the Assembly Language translated by the C/C++ compiler

Hydra Card Game | C/C++

(Private Repository, message to see)

- Developed the backend of a computerized card game using Object-Oriented Programming and advanced C/C++ concepts such as smart pointers, inheritance, polymorphism
- Managed memory on the heap using smart pointers improving efficiency
- Implemented a user friendly MVC (Model View Controller) design pattern that promoted high cohesion, low coupling and the concept of encapsulation and information hiding