ABHI PATEL

📱+1(267)977-6054 ✉️ ap2566@scarletmail.rutgers.edu 🔗abhipatel.in 👾abhipatel2810📍NJ, USA.

Summary:

Results-driven Software Developer with a strong foundation in full-stack development, cloud computing and scalable web applications. Currently pursuing a master's in computer science at Rutgers University, I specialize in ReactJS, Next.js, TypeScript and CSS for modern frontend development, while leveraging Google Cloud Platform (GCP),AWS, and Firebase for backend scalability. With hands-on experience in database architecture, API integrations and CI/CD pipelines, I am adept at optimizing application performance and delivering efficient, user-centric solutions. Passionate about solving complex technical challenges, I thrive in fast-paced environments where I can drive innovation, automation, and system efficiency through cloud-native and AI-driven technologies.

Education:

Bachelor's in Computer Engineering (Jun 2019 - May 2023) **CPI : 8.16/10**
Sarvajanik College of Engineering & Technology, Surat Gujarat Technological University, Gujarat
Master in Computer Science (Jan 2024) **GPA : 3.4/4.0**
Rutgers University— New Brunswick

Academic Projects & Research:

Designed and analyzed algorithms for logic resolution in SAT-to-SCC graph transformations.
Conducted comparative benchmarking of classical vs parallel machine learning models for prediction tasks.
Published GitHub-hosted whitepaper with detailed README on model performance, available at: github.com/abhipatel2810/forest-fire-svm.
Repo Link: github.com/abhipatel2810/empath-r-ai-support (includes README and deployment logs)
Keywords: Theoretical Computer Science, Graph Theory, Automata, Complexity, Formal Verification, Proof of Correctness, Whitepaper Author.

Relevant Coursework:

Operating Systems, Algorithms, Data Structures, Time Complexity, Distributed Systems, Compiler Design, Theory of Computation.

Experience:

Adaptable Service, 07/2019 – 01/2024
I collaborated with a 24+ member technical team, delivering 28+ projects involving scalable web applications, dynamic websites, and static websites. I developed and optimized ReactJS, Next.js, and TypeScript-based frontends, integrating RESTful and GraphQL APIs for seamless data exchange. I extensively worked with Google Cloud Platform (GCP) services, leveraging Cloud Functions, Firestore, BigQuery and Cloud Run to build scalable and efficient cloud-based applications. I also implemented CI/CD pipelines with Google Cloud Build, improving deployment efficiency. Additionally, I integrated Google Maps API, Google Analytics, and Google Authentication to enhance web applications with location intelligence, data insights, and secure user management. My contributions in performance optimization, caching strategies, and database indexing helped reduce application load times by 40%, improving user experience and system efficiency.Worked in Agile teams using JIRA and Asana. Followed TDD principles and implemented unit testing using PyTest and Mocha. Participated in weekly sprint planning and code reviews.
Software Engineering: Unit Testing (PyTest, Mocha), Agile & Scrum, SOLID Principles, MVC Architecture, SDLC, Version Control (Git, GitHub), Code Reviews, Design Patterns.
GGate, 12/2021 – 06/2022 (Part-Time)
I worked as a database specialist, specializing in database management and architecture. I designed and optimized relational (MySQL, SQLite) and NoSQL (Firebase) databases, implementing N-N structured and hierarchical data models to improve query efficiency by 35%. I developed and maintained ETL pipelines for seamless data ingestion and transformation, ensuring high availability and scalability in serverless architectures. Additionally, I enhanced data integrity and security by enforcing ACID compliance, indexing strategies, and role-based access control (RBAC) to optimize data retrieval and storage operations.

Projects:

Google Play Store Ratings Analysis: Leveraging Python (Pandas, Matplotlib, Seaborn) for exploratory data analysis (EDA), this project utilized Tableau and Looker for advanced visualizations to identify trends, correlations, and anomalies in app ratings. By providing a data-driven framework, it enhances decision-making for app optimization, improving overall user

satisfaction.Applied data cleaning techniques and created dashboards in Tableau and Looker with KPIs including daily active users and churn rate. Conducted trend forecasting and anomaly detection.
Tech Stack: Python (Pandas, Matplotlib, Seaborn), Tableau, Looker.
Data Analysis: SQL Query Optimization, Excel (Pivot Tables, VLOOKUP), Data Cleaning, KPI Dashboards, PowerBI, A/B Testing, Trend Forecasting, ETL Process Management.

Medical Care - Patient to Doctor Services: This project involved developing a web application using PHP, SQL, JavaScript, and Bootstrap for secure patient data management. It streamlined healthcare processes by integrating features like medicine reminders, invoice generation, and appointment scheduling, offering real-time access to medical records for better emergency response and patient care efficiency.
Tech Stack: PHP, SQL, JavaScript, HTML, CSS, Bootstrap 5.

Forest Fire Detection: Implementing a Parallel Support Vector Machine (SVM) model using Python (NumPy, Pandas), this system processed large datasets in parallel to predict forest fires. By analyzing fire patterns efficiently through parallel computing, it supports early fire detection, enabling faster responses to prevent large-scale disasters.
Tech Stack: Python (Parallel SVM, NumPy, Pandas), Database Management, and Parallel Computing.

Survey Dashboard Creation: This project built an interactive dashboard utilizing Google Charts and Python (Flask) with REST API integration, enabling real-time data visualization and analytics. The dashboard supports dynamic user interaction and visualization of complex datasets, offering a powerful tool for businesses to monitor and analyze performance metrics.Focused on responsive design, cross-browser compatibility, and Web Accessibility (WCAG 2.1). Tested frontend components using Jest and integrated UI designs from Figma.
Tech Stack: Google Charts, Python (Flask), REST API, HTML, CSS, JavaScript.
Frontend Testing & Design: Jest, Cypress, Lighthouse, Responsive Design, SEO Optimization, WCAG Accessibility, Figma, Adobe XD.

2SAT-to-SCC Conversion and 3D Graph Representation: Using Python (NetworkX, Matplotlib) and 3D graph libraries, this project developed an algorithm to convert 2SAT clauses into Strongly Connected Components (SCCs). SCCs that were made could be seen in 3D, which made it possible to interact with them and look at complicated logical structures and network connections. This capability was useful for both computational theory and network analysis.
Tech Stack: Python (NetworkX, Matplotlib), JSON, 3D graph libraries, HTML, CSS, and JavaScript.

Cloud-Based Microservices Deployment with Kubernetes: Developed a scalable microservices architecture using Kubernetes for container orchestration, ensuring high availability and load balancing. Deployed multiple microservices with Docker and Kubernetes, leveraging Helm charts for efficient service management and automated rollouts. Implemented CI/CD pipelines with Google Cloud Build to automate deployments and enhance system reliability. Optimized resource allocation using Horizontal Pod Autoscaler (HPA) and Kubernetes Ingress for efficient traffic management. Ensured secure communication between services using RBAC and Kubernetes Secrets, reducing security vulnerabilities.
Tech Stack: Kubernetes, Docker, Google Cloud Build, Helm, CI/CD, Python, Flask, ReactJS.

Intelligent DNS Resolver with ML-Based Upstream Selection: Designed and implemented a smart DNS resolver in Python that integrates a machine learning engine to dynamically select the optimal upstream DNS server (Google, Cloudflare, Quad9, ISP) based on real-time performance metrics such as RTT, cache hit ratio, and error rates. Incorporated Prometheus and Grafana for metrics collection and visualization, and containerized the full system using Docker Compose. Load testing was conducted using Locust for performance benchmarking under varied query patterns.Secured REST APIs with JWT authentication and integrated OAuth 2.0 for third-party access. Enabled logging using ELK stack and added Redis caching for performance optimization.
Tech Stack: Python, Flask, Scikit-learn, Prometheus, Grafana, Docker, Locust, DNS Protocol, REST API.

EMPATH-R: Emotionally Adaptive Mental Health Companion : Developed a socially cognizant chatbot for elderly mental health support using OpenAI's GPT-3.5 and facial emotion recognition. Integrated multimodal emotion detection (textual + visual) using the FER library and GPT prompting to adapt conversational tone in real time. Implemented ethical filters, risk detection (e.g., suicide/hurt terms), and mood logging to ensure safe interaction. Emphasized Theory of Mind and responsible AI design in emotionally vulnerable contexts.
Tech Stack: Python, OpenAI GPT-3.5 APIn or Ollama API or local level intregation , FER (Facial Emotion Recognition), Gradio, CSV Logging, Ethical AI Design

Backend Engineering: JWT, OAuth 2.0, Load Balancing, API Gateway, Logging (ELK Stack), Monitoring (Grafana, Prometheus), Caching (Redis), Rate Limiting, Circuit Breakers.

Education to Employment: Data Mining for Job Offers and Salary Prediction : Conducted an end-to-end data mining and machine learning project on a dataset of 5,000 students to predict (1) the number of job offers and (2) starting salary. Employed regression models (Linear, XGBoost, Random Forest) and classification models (XGBoost, Random Forest, Logistic Regression), using SHAP values for interpretability. Engineered derived features (e.g., Skill Index, Certification Density), handled class imbalance with SMOTE, and ensured robust evaluation using cross-validation and error analysis. Delivered insights to optimize academic advising and career planning for students.Implemented model pipelines using Scikit-learn and XGBoost; used SMOTE for class imbalance handling. Feature importance analyzed using SHAP values and ROC-AUC metrics.

Tech Stack: Python (scikit-learn, XGBoost, SHAP), Pandas, Matplotlib, Seaborn, Google Colab, Git.

Data Science Tools: Jupyter Notebooks, Airflow (for scheduling), MLflow (for model tracking), Feature Engineering, Precision/Recall, ROC-AUC, Model Validation, Cross-Validation, A/B Testing.

Big Data: PySpark, Hadoop (basic knowledge), Vertex AI (GCP), Amazon SageMaker.

Education to Employment: Data Mining for Job Offers and Salary Prediction : Conducted an end-to-end data mining and machine learning project on a dataset of 5,000 students to predict (1) the number of job offers and (2) starting salary. Employed regression models (Linear, XGBoost, Random Forest) and classification models (XGBoost, Random Forest, Logistic Regression), using SHAP values for interpretability. Engineered derived features (e.g., Skill Index, Certification Density), handled class imbalance with SMOTE, and ensured robust evaluation using cross-validation and error analysis. Delivered insights to optimize academic advising and career planning for students.Implemented model pipelines using Scikit-learn and XGBoost; used SMOTE for class imbalance handling. Feature importance analyzed using SHAP values and ROC-AUC metrics.

Tech Stack: Python (scikit-learn, XGBoost, SHAP), Pandas, Matplotlib, Seaborn, Google Colab, Git.

Data Science Tools: Jupyter Notebooks, Airflow (for scheduling), MLflow (for model tracking), Feature Engineering, Precision/Recall, ROC-AUC, Model Validation, Cross-Validation, A/B Testing.

Big Data: PySpark, Hadoop (basic knowledge), Vertex AI (GCP), Amazon SageMaker.

Skill Set:

Frontend:

ReactJS, Redux, Next.js, TypeScript, Tailwind CSS, Bootstrap 5, HTML, CSS, RESTful APIs, GraphQL, WebSockets

Backend & Cloud with languages:

Python, Golang, C, JAVA, JavaScript, Node.js, Flask, PHP, AWS (Lambda, S3, DynamoDB, EC2), Firebase, Google Cloud (GCP)

SQL (MySQL, PostgreSQL), NoSQL (MongoDB, Firebase Firestore)

Library & Tools & Platforms:

Git, Docker, JIRA, Asana - CI/CD Pipelines

NetworkX, Pandas, Matplotlib, Google Charts

Tableau, Apache Superset, Looker

Licenses & certifications:

Google Cloud Fundamentals: Core Infrastructure

Google Cloud Big Data and Machine Learning Fundamentals

Python for Data science and machine learning bootcamp

Google Workspace Sales Credential & Google Cloud Sales Credentials