Jinesh Patel

jmpatel2113@gmail.com | (973)987-2160 | Somerset, NJ 08873
linkedin.com/in/jineshpatel02 | github.com/jmpatel2113 | jmpatel2113.github.io/jineshpatel

Education

Rutgers University | New Brunswick, New Jersey — Anticipated Graduation: January 2026

Bachelor of Science (B.S.) | Major: **Computer Science** | Minor: **Business Administration**

GPA: 3.7, Relevant Courses: Data Structures(**Java**), Software Methodology(**Java**), Data Management & Science(**Python, SQL**)

AWS Cloud Institute | Cloud Application Development Track — Anticipated Completion: April 2026

Relevant Course: AI for Developers, Cloud Fundamentals and Operations, Developer Intermediate(**Python, Javascript, Django**)

Skills

Languages: Python, Java, Javascript, C, HTML, CSS (Tailwind), SQL

Certifications: AWS Certified Cloud Practitioner

Tools & Frameworks: React, Django, Node.js, Typescript, MongoDB, Git, Express.js, Firebase, ServiceNow, Keras, Scikit-learn

AWS Services & Tools: Lambda, DynamoDB, IAM, S3, VPC, Subnets, Cloudwatch, Rekognition, Textract, API Gateway

Experiences

Tuvalabs Inc., *Software Engineer Intern* — New York, NY, 06/2025 - 08/2025

- Contributed to building dynamic dashboard using **React** & **FastAPI** to develop frontend components & backend APIs
- Participated in code reviews and wrote unit tests to improve software quality & stability while following agile **SDLC**
- Assisted in deploying the web application on **AWS**, gaining hands-on experience with cloud services and technologies

Thaddeus, *Software Engineer Intern* — Remote, 11/2024 - Present

- Refactored internship & volunteer application webpages with **React & Typescript**, improving user engagement by 15%
- Built an HR-managed employee component that streamlined staff visibility, increasing profile completeness by 30%
- Solved deployment failures by updating Dockerfile and authentication keys, ensuring CI/CD pipeline stability

Headstarter AI, *Software Developer Fellow* — Remote, 07/2024 - 09/2024

- Developed multiple full stack projects in tight weekly Agile sprints while collaborating with a 3-member dev team
- Utilized **React, MongoDB and Gemini AI API** frameworks, including Google Analytics for user traffic and engagement
- Built **Mock Mentor**, an AI-powered mock interview service used by 7+ users for dynamic questions & instant feedback

Rutgers University OIT, *Student IT Support* — New Brunswick, NJ, 04/2023 - 01/2024

- Provided clear, empathetic IT support and reduced queue length by 20% while documenting & updating ticket details
- Resolved 8-10 technical support ticket/shift via phone, email & in-person using **ServiceNow** at an 85% resolution rate
- Assisted in handling university-wide outages by coordinating with senior IT staff & relaying updates to impacted users

Projects

AWS Cloud Capstone I | Bank Customer Onboarding KYC Application

- Built a serverless KYC app architecture with **Lambda, IAM, SAM** and **S3** for automated document verification
- Automated end-to-end KYC workflow using **Step Functions**, reducing manual processing time & meeting compliance
- Integrated **DynamoDB, Rekognition** & **Textract** for secure data store in database, facial comparison & data extraction

Mock Mentor | AI-Generated Mock Interview platform

- Enabled 7 users to generate 35+ AI-led questions by leveraging **Gemini AI** for question generation & feedback analysis
- Automated end-to-end AI evaluation using prompt engineering & speech-to-text, improving feedback accuracy
- Integrated **NextJs & Tailwind CSS** to build robust frontend, **Node.js** for backend and Drizzle ORM-Neon database

REST API | User authentication API

- Designed REST API supporting CRUD operations on user accounts using **Express, Typescript, MongoDB** & **Mongoose**
- Implemented user authentication & access control via registration/login endpoints, following basic security practices

Stock Trend Prediction Model | Machine Learning Stock Trend Model

- Built a streamlit app in **Python** that forecasts stock trends using LSTM model developed with **Keras** and **TensorFlow**
- Utilized **pandas, scikit-learn and matplotlib** for data cleaning/preparation, normalizing data and visualizing results
- Applied 100-Day and 200-Day moving averages indicators and 70-30 train-test data split for model evaluation