Dhruvish Patel

Software Developer

My Contact

✉ pateldhruvish612000@gmail.com

☎ +91 6353573222

in https://www.linkedin.com/in/dhruvish09/

https://github.com/Dhruvish09

Hard Skill

- BackEnd: Python
- Framework: FastAPI, Django, Flask
- DB: MongoDB, MySQL, Redis, Pinecone
- Web Crawling: BeautifulSoup, Crawler4AI
- AI/LLM: Model Integration
- Testing & QA: Pytest, Nose, Locust
- FrontEnd: HTML, CSS, Bootstrap
- DevOps: AWS, Firebase, Docker, Github Action.
- API Documentation: Postman, Swagger
- Version Control & Collaboration: GitHub, Jira

Soft Skill

- Observation
- Communication
- Multi-tasking

Education Background

- sardar patel collage of engineering(2021)
  Bachelor's in Information & technologies
  Scored 8.24 CGPA
- Government Polytechnic Ahmedabad(2018)
  Diploma in Information & technologies
  Scored 8.90 CGPA
- S.S.C (2015)
  Scored 95%

About Me

Experienced python developer with nearly 4 years of expertise in building scalable, secure backend systems and AI-powered platforms. Passionate about staying up-to-date with the latest advancements in the field. I am Committed to delivering high-quality code and contributing to the success of projects through effective teamwork and communication.

Professional Experience

Weam AI | Software Developer
2024 – Present

Key responsibilities:

- Developed scalable backend services using FastAPI.
- Integrated LLM APIs (OpenAI, Claude, Gemini, HuggingFace) with LangChain for AI workflows.
- Built RESTful APIs with JWT, CSRF protection, and asynchronous task queues using Celery and Redis
- Implemented web scraping and file parsing tools using Crawler4ai Playwright and BeautifulSoup
- Wrote automated tests using Pytest and performed load test with Locust
- Deployed applications with Docker, Docker Compose.

Zymr System | Software Developer

2021 – 2024

Key responsibilities:

- Analyze Requirements and provide solutions
- Create REST APIs with middleware in python.
- Create a wide variety of unit tests to verify the functionality of software.
- Develop solutions following industry best practices.
- Follow Agile methodologies and complete targets for sprints.
- Design Database according to requirements.

Achievements

2013 – 2015

Honorable Jaydeep Sinh Baria Award – Government of India

Received this prestigious award for winning the State Netball Championship three consecutive times and representing Gujarat at the National Level Netball Tournament.

Projects

WEAM AI Platform
2024-Present

Domain:

- AI-Powered Communication, Agents & Automation Platform

Project Overview:

- Weam AI is a collaborative AI workspace that unifies leading language models like ChatGPT, Claude, Gemini, and LLaMA into a single platform. It empowers teams to build and manage custom AI agents, organize reusable prompt libraries, and collaborate in real time across document processing, chat workflows, and SEO tasks — all without the need for multiple tool subscriptions.

Architecture:

- Microservices architecture using FastAPI and Celery for scalable backend operations.
- Integrated LangChain and Pinecone for prompt orchestration and vector storage.
- Redis + MongoDB used for caching and metadata storage.
- Real-time insights and logs handled via Prometheus, Grafana, and Loki.
- Secure API gateway built with JWT, CSRF protection, and streaming response support.

Contributions:

- Developed LLM-based backend workflows for agent creation, embedding, chat, and scraping.
- Integrated multiple AI providers (OpenAI, Gemini, Anthropic, HuggingFace) with dynamic routing.
- Built background task queues for document processing, video/audio analysis, and S3 syncing.
- Led API development for secure and scalable model interaction endpoints.
- Supported frontend team with REST APIs and templated responses for agent and prompt UI.

Tech Stack:

- Backend: Python, FastAPI
- AI & LLM Integration: OpenAI API, Anthropic (Claude), Gemini, Hugging Face
- Databases & Storage: MongoDB, Pinecone, AWS S3, Redis
- Web Crawling & Content Extraction: Crawler4AI, Playwright, BeautifulSoup
- Task Processing & Automation: Celery (with Redis broker), Playwright, Crwaler4AI
- Monitoring & Logging: Prometheus, Grafana, Loki
- Security: JWT, CSRF protection, cryptography libraries[Encryption & Decryption]
- DevOps & Deployment: EC2, ECR, S3, Codepipeline, Cloudwatch, Docker, Docker Compose, Github Actions
- Templates: Bootstrap, CSS, Jinja2
- Testing & QA: Pytest, Locust
- Notifications: Firebase Cloud Messaging (FCM), SMTP/SES (email alerts)

ZQA (Integrated Project Insights Management)
2024-2024

Domain:

- Project Management Analytics & Engineering Insights

Project Overview:

- A project intelligence platform designed to centralize and visualize engineering data from tools like Jira, GitLab, Jenkins, SonarQube, and Zephyr. The system provides real-time dashboards for tracking test coverage, deployment health, and delivery velocity—helping teams make data-driven decisions and optimize DevOps workflows.

Architecture:

- Monolithic backend built with Python (Flask) for REST APIs, data aggregation, and transformation.
- Data ingested from third-party systems via APIs and stored securely in MySQL.
- Integrated with DevOps and QA tools (Jenkins, Zephyr, SonarQube) for complete CI/CD visibility.
- Interactive dashboards built in ReactJS present real-time insights into project and engineering performance.

Contributions:

- Developed core backend services using Flask for multi-source data extraction and transformation.
- Designed database schema and implemented secure MySQL queries for high-volume, structured data storage.
- Created RESTful endpoints to serve KPIs and metrics to the ReactJS frontend.
- Built reusable logic for parsing and correlating deployment logs, test results, code coverage, and issue tracking data.
- Collaborated with DevOps and QA teams to integrate CI/CD pipelines and QA reporting tools into the analytics workflow.

Tech Stack:

- Backend: Python (Flask)
- Databases: MySQL
- Frontend: ReactJS (team-owned), HTML, CSS, Bootstrap
- CI/CD & DevOps: Jenkins, Docker
- Integrations: Jira, GitLab, Zephyr, Jenkins, SonarQube
- Collaboration & Version Control: GitHub

Portfolio builder

2022 – 2023

Domain:

- Web Portal – Dynamic Portfolio Builder

Project Overview:

- A responsive web platform designed to help users dynamically build and manage their personal or professional portfolios. The system allows users to customize sections, upload project content, and showcase their skills with a modern, interactive UI.

Architecture:

- Developed with Django as the backend framework for dynamic content rendering and user management.
- Frontend designed using Bootstrap, with responsive layout and reusable components.
- Secure user authentication integrated with email verification.
- Media and document uploads handled via AWS S3 for scalable storage.
- Background operations such as image optimization and email notifications handled via Celery + Redis.
- Containerized using Docker for consistent deployment across environments.

Contributions:

- Built and deployed a feature-rich portfolio builder using Python (Django) with support for custom sections and templates.
- Integrated MySQL for persistent user data, project content, and authentication metadata.
- Implemented email-based user registration and verification workflows.
- Managed file storage and media delivery using AWS S3 integration.
- Set up Dockerized development and production environments for seamless deployment.
- Integrated Celery workers to handle async tasks like sending confirmation emails and background jobs.

Tech Stack:

- Backend: Python(Django)
- Databases: MYSQL, Redis
- Frontend: HTML, CSS, Bootstrap
- DevOps & Storage: Docker, AWS S3
- Task Processing: Celery, Redis
- Authentication & Emailing: SMTP, Email verification workflows
- Version Control & Tools: Git

Komplyd

2021 – 2022

Domain:

- SaaS Product – Cannabis Compliance & Enterprise Data Platform

Project Overview:

- Komplyd is a cloud-native compliance platform designed for the cannabis industry. It offers a robust API layer and enterprise data platform to help businesses navigate regulatory requirements, integrate compliance workflows, and manage state-by-state operational data securely and efficiently.

Architecture:

- Built a secure REST API layer using Python and Flask for scalable, low-maintenance backend operations.
- Integrated with MySQL for storing regulatory data, license metadata, and operational records.
- APIs followed industry-standard practices for authentication, data validation, and error handling.
- Designed to support multi-tenant deployments across regions with cloud-based scalability.

Contributions:

- Developed and maintained Python-based REST APIs using Flask for interacting with compliance data and third-party services.
- Implemented secure CRUD operations and validations for licensing, compliance checks, and business operations.
- Wrote unit and integration tests using Nose to ensure API correctness and performance.
- Contributed to version control and CI workflows using Git, collaborating with cross-functional cloud teams.

Tech Stack:

- Backend: Python(Flask)
- Databases: MYSQL
- Testing & QA: Nose
- Version Control & Tools: Git