

Prateek Yadav

Software Engineer Intern

Summary

Software Engineer Intern with experience in Python, backend development, and full stack application development. Currently working at Grid Dynamics using Python, FastAPI, Flask, Git, and Docker. Built projects including a log analyzing tool, a mail reminder application, a Trello style backend API, and AWCP, a multi agent system using FastAPI, Temporal, OpenTelemetry, and React. Strong interest in building clean and reliable software and learning modern technologies.

Technical skills

- **Programming Languages:** Python, JavaScript, HTML, CSS
- **Backend:** FastAPI, Flask
- **Databases:** PostgreSQL, MySQL
- **ORM and Migrations:** SQLAlchemy, Alembic
- **Developer Tools:** Git, Docker, VS Code
- **Workflow and Observability:** Temporal, OpenTelemetry
- **Issue Tracking:** Jira
- **Platforms:** macOS, Windows

Education and Certificates

- 2022 – 2024, Master of Computer Applications (MCA), AKTU Lucknow
- 2019 – 2022, Bachelor of Science (B.Sc.), CSJMU Kanpur
- 2025, The Complete Python Bootcamp From Zero to Hero in Python, Udemy

Languages

- English - B1
- Hindi - Native Speaker

EXPERIENCE

Grid Dynamics , Software Engineer Intern
{Location-Hyderabad ,India}
January 2026 – Current

Projects

Log Analyzing Tool

A command line tool built in Python that reads a server log file and shows useful statistics. The tool reads the file line by line, so it can handle very large log files without using too much memory.

Responsibilities:

- Read log files line by line to keep memory usage low.
- Used regular expressions to get IP addresses from log files.
- Counted requests for each IP address and endpoint.
- Calculated total requests, success responses, client errors, and server errors.
- Displayed the most active IP addresses and most used endpoints.
- Built a simple command line menu for different reports.

Technologies: Python, Regular Expressions, File Handling, CLI

Mail Reminder Application

A reminder application built with a desktop interface, a Flask API, and an email sender. Users can create reminders and receive emails on the selected date and time.

Responsibilities:

- Built the desktop application using PyQt6.
- Created a Flask REST API to save reminder data.
- Stored reminder data in CSV files.
- Sent reminder emails using Gmail SMTP.
- Added daily, weekly, and monthly repeat reminders.
- Organized the project into controllers, services, and utility modules.

Technologies: Python, Flask, PyQt6, smtplib, CSV

Trello App Backend REST API

A backend REST API for a Trello style project management application built with FastAPI and PostgreSQL. The application manages users, boards, sections, tickets, and board invitations.

Responsibilities:

- Built REST API endpoints using FastAPI.
- Added user registration and login with JWT authentication.
- Protected private APIs with bearer token authentication.
- Stored passwords securely using passlib and bcrypt.
- Created database models using SQLAlchemy.
- Managed database changes using Alembic.
- Built a board invitation system with one time invite tokens.
- Added create, read, update, and delete operations for sections and tickets.
- Organized the project into routers, services, models, and schemas.
- Used Docker Compose to run the API and PostgreSQL together.

Technologies: Python, FastAPI, PostgreSQL, SQLAlchemy, Alembic, JWT, passlib, Docker

AWCP - Agent Workforce Control Plane (Capstone Project, Current)

A governed multi agent system that manages many AI agents. It uses FastAPI for backend services, Temporal for workflow management, OpenTelemetry for observability, and React for the web dashboard.

Responsibilities:

- Built FastAPI services for the gateway and control APIs.
- Started agent workflows and tracked their status.
- Created Temporal workflows and activities for each agent step.
- Used Temporal signals to update running workflows with live events.
- Added OpenTelemetry for traces, metrics, and logs.
- Instrumented FastAPI APIs and HTTP requests.
- Added custom metrics for workflows, activities, tools, and LLM calls.
- Connected telemetry with Grafana, Prometheus, Loki, and Tempo.
- Built the React dashboard with live workflow and token monitoring.
- Used Tailwind CSS to build the frontend interface.

Technologies: Python, FastAPI, Temporal, OpenTelemetry, React, Tailwind CSS, Grafana, Prometheus, Docker