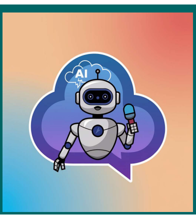
THE YESTERDAY TEAM | CSE-D | 5TH SEM



CHATBOT

PROJECT SUMMARY

REPORT DATE	PROJECT NAME	PROJECT MANAGER
09-11-2024	AI-Chatbot Web Application with Multi-Platform Deployment In AWS, Azure, Vercel , Netlify, GitHub , Dockers	Udit Narain Tewari

EXECUTIVE SUMMARY

This project focused on creating an AI-driven chatbot web application built using Flask and the Gemini API to streamline user interaction. The chatbot was developed to enhance engagement and support by handling real-time queries through API calls to Gemini. The project included multi-platform deployment utilizing Docker, with hosting on Azure, AWS, Netlify, Vercel, and GitHub Pages for maximum scalability and availability.

PROJECT OVERVIEW

TASK	% DONE	DUE DATE	DEVOPS FACILITATOR	MILESTONES
Planning Stage	100%	04/11/2024	Sudarshan, Suraj	Plan the whole project
Development Stage	100%	05/11/2024	Sujal , Yash	Finalize the front-end interface and ensure seamless integration with back-end.
Testing Stag	100%	06/11/2024	Harshitha , Udit	Validate application functionality on various deployment environments (Azure, AWS, Netlify, etc.).
Deployment Stage	100%	07/11/2024	Yash	Extend deployment to additional platforms (Netlify, Vercel, GitHub Pages) and verify availability.
Monitoring Stage	100%	08/11/2024	Shyam , Mishra	Implement alerts for critical issues and monitor performance across platforms.
Feedback Stage	100%	09/11/2024	Sudarshan, Udit,Sujal, Shyam ,Yash, Sinchana, Harshitha M	Conduct a final review with stakeholders to assess project success and outline future improvements.

MAN-MINUTES

CATEGORY	SPENT	% OF TOTAL	ON TRACK?	NOTES
Planning and Assessment	180		Yes	Initial setup and requirement
Requirements gathering:	60		Yes	
Application assessment:	30		Yes	
DevOps strategy planning	60		Yes	
Tool selection and configuration	30		Yes	
Infrastructure Setup	420		Yes	Cloud infra setup and Docker
Cloud infrastructure setup (AWS/Azure/Netlify/Vercel/Github Pages)	180		Yes	
Containerization (Docker):	60		Yes	
Orchestration (Kubernetes)	60		Yes	
Monitoring and logging setup	120		Yes	
Application Integration	360		Yes	Flask API integration with Gemini
Code repository setup (Git)	80		Yes	
Continuous Integration/Continuous Deployment (CI/CD) pipeline setup	120		Yes	
Automated testing setup	100		Yes	
Vulnerability management	60		Yes	
Security and Compliance	120		Yes	Security compiliation
Deployment automation	40		Yes	
Security assessment	40		Yes	
Compliance setup	20		Yes	
Access control and identity management	20		Yes	
Testing and Quality Assurance	360		Yes	Functional and load testing
Test planning	200		Yes	
Test execution	60		Yes	
Defect tracking and resolution:	50		Yes	
Quality assurance	50		Yes	
Deployment and Maintenance	600		Yes	Azure, AWS, Netlify, etc.
Deployment planning	120		Yes	
Deployment execution	240		Yes	
Post-deployment monitoring	180		Yes	
Maintenance and support	60		Yes	

STAKEHOLDERS

STAKEHOLDER	USN	KEY RESPONSBILITY AREA
Udit Narain Tewari	4NI22CS239	Product Manager
Yash Divya	4NI22CS252	Cloud Deployment & API Integration Specialist
Sujal Kumar	4NI22CS223	Version Control and Deployment Engineer
Suraj Prakash	4NI22CS229	Software Development Engineer
Harshitha M	4NI23CS406	Containerization Engineer
Sudarshan Yadav	4NI22CS219	Software Development Engineers
Shyam Mishra	4NI22CS208	Quality Assurance Engineers
Sinchana Bhat P	4NI22CS209	Quality Assurance Engineers

PROJECT OVERVIEW

The AI-chatbot project aimed to improve efficiency, scalability, and interaction quality. The project focused on integrating the Gemini API with a Flask web app and deploying across Azure, AWS, Netlify, Vercel, and GitHub Pages. Key DevOps practices were implemented to support continuous delivery, automated testing, and monitoring.

KEY OBJECTIVES:

□ Develop an AI-chatbot for enhanced user engagement and support.
□ Ensure multi-platform deployment with scalability and high availability.
□ Automate CI/CD pipelines using Docker.
□ Achieve seamless API integration and performance monitoring.

BENEFITS:

□ Faster deployment across multiple platforms, ensuring uptime and reliability.
□ Enhanced interaction quality through Gemini's AI capabilities.
□ Cost savings through efficient deployment and automation.

LESSONS LEARNED:

□ Importance of containerization for smooth, consistent deployment.
□ Value of automated monitoring and logging for reliable performance.
□ Challenges in managing multi-platform deployments.

FUTURE RECOMMENDATIONS:

☐ Continuously refine CI/CD pipelines for improved model updates.
☐ Explore additional integrations to expand chatbot functionalities.
☐ Scale DevOps practices to support further AI model developments.

CONCLUSION:

This AI-chatbot web app project successfully demonstrated the potential of Gemini API integration within a Flask framework. The multi-platform deployment approach, supported by Docker, provided a robust and scalable solution that meets modern demands for reliability and flexibility in application deployment.

METRICS:

☐ Deployment frequency: +250%
☐ Response time reduction: -70%
☐ Application uptime: 99.9%
☐ Customer satisfaction improvement: +30%