INNOVATE_HERS | CSE-D | 5TH SEM

COMMUNITY SUPPORT SYSTEM

PROJECT SUMMARY

REPORT DATE	PROJECT NAME	PROJECT MANAGER
09-11-2024	Community support Deployment In AWS, Netlify,GitHub,Dockers	Shikhi Chandrashekar

EXECUTIVE SUMMARY

To get started right away, just tap any placeholder text (such as this) and start typing to replace it with your own.

PROJECT OVERVIEW

TASK	% DONE	DUE DATE	DEVOPS FACILITATOR	MILESTONES
Planning Stage	100%	04/11/2024	Shikhi, Sreelakshmi	Plan the whole project
Development Stage	100%	05/11/2024	Umema , sindhu,Brunda	Finalize the front-end interface and ensure seamless integration with back-end.
Testing Stag	100%	06/11/2024	Yashaswi , Vismaya	Validate application functionality on various deployment environments (Azure, AWS, Netlify, etc.).
Deployment Stage	100%	07/11/2024	Brunda , sindhu,Yashaswi	Extend deployment to additional platforms (Netlify, Vercel, GitHub Pages) and verify availability.
Monitoring Stage	100%	08/11/2024	Umema, Sindhu , Shikhi	Implement alerts for critical issues and monitor performance across platforms
Feedback Stage	100%	09/11/2024	Yashaswi , Vismaya , shikhi , umema , sindhu , brunda , sreelakshmi	Conduct a final review with stakeholders to assess project success and outline future improvements.

MAN-HOURS

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/GCP)	180		YES	
Containerization (Docker):	60		YES	
Orchestration (Kubernetes)	60		YES	
Monitoring and logging setup	110		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
Shikha Chandrashekar	4NI22CS198	Product Manager
Umema Zuha	4NI22CS240	Software Development & AI tools
Yashaswi S S	4NI23CS253	Cloud Deployment & API Integration

Vismaya R	4NI22CS249	Version Control and Deployment Engineer
V Brunda	4NI22CS242	Containerization Engineer
Sindhu M R	4NI22CS211	Frontend Developer
Sreelakshmi Nair	4NI22CS261	Backend Developer

PROJECT OVERVIEW

The DevOps project aimed to improve the efficiency, reliability, and scalability of [Company Name]'s software development and deployment processes. The project focused on implementing DevOps practices, automating CI/CD pipelines, and ensuring continuous monitoring and feedback.

KEY OBJECTIVES:

□ Efficient Support: Quick, accurate AI responses with human escalation as needed.
□ Engagement: Discussion boards, user posts, and rewards for active members.
□ Scalability: Cloud-based, high availability for peak usage.
□ Automation: CI/CD pipelines, Docker for deployments, automated monitoring.
□ Analytics: Track performance, user satisfaction, and improvement insights.

BENEFITS:

□ User Satisfaction: Quick, accurate support enhances trust.
□ Increased Engagement: Active forums and rewards boost community involvement.
□ Reliable Access: High availability supports users anytime.
□ Efficiency: Automation saves time and simplifies updates.
□ Wider Reach: Integrations enable multi-platform support.

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:

In conclusion, a well-designed community support system that prioritizes efficient user support, engagement, scalability, and data security delivers substantial value. By leveraging automation, multi-platform integration, and insightful analytics, such a system not only enhances user satisfaction but also builds a thriving, engaged community. These objectives ensure the platform remains responsive, reliable, and adaptable to future growth, fostering a supportive environment where users feel valued and connected.

METRICS:

- Deployment frequency: +250%
- Response time reduction: -60%
- Application uptime: 99%
- Customer satisfaction improvement: +30%