Azure migration costs calculator with embedded AI

Project Plan

CS 4850 - Section 02 – Fall 2024

Aug 21, 2024

Roles	Name	Major responsibilities	Contact
Project owner	Capstone		
Team leader	Kunal Shenoi	Organize meetings, monitor progress, and submit deliverables. Develop and implement testing protocols.	678-779-4770
Team members	Graham Allen	Developer - Developing back-end AI integration, implementing Azure/AWS API calls, and database connections. Assisting in front-end UI/UX development	770-841-6718
	Angel Hernandez	Developer - Designing and implementing a user-friendly interface, while assisting with backend and AI integration and development	770-318-5359
	Yvan Ngah	Documentation - Creating, maintaining, and ensuring the accuracy of all technical documentation.	171haden@gmail.com
Advisor / Instructor	Sharon Perry	Facilitate project progress; advise on project planning and management.	770-329-3895

Kunal Shenoi
Team Leader, Test

Graham Allen

Developer

Angel Hernandez
Developer

Yvan Ngah
Documentation

Table of Contents

1.0 Project Overview / Abstract (Research)

The project aims to develop a web-based Migration Cost Calculator with Embedded AI that assists in evaluating the cost-benefit of migrating from on-premises infrastructure to Azure cloud services. The application will utilize AI language analysis to understand user-provided details on infrastructure and return detailed cost analysis reports by calling Azure Pricing APIs. The project scope includes assessing current on-premises infrastructure, designing and developing the web application, integrating AI language analysis and Azure pricing APIs, and conducting comprehensive testing and analysis. The final deliverable will be a fully functional web application, along with project documentation and recommendations.

2.0 Project website

website.com

Deliverables -

Requirements

A. Meet with Stakeholder(s): Initial meeting to gather project requirements.
B. Define Requirements: Create a comprehensive list of project requirements.
C. Review Requirements with Stakeholder(s): Ensure alignment on project goals.
D. Get Sign-off on Requirements: Obtain formal approval from all stakeholders.

Project Design

A. Define Tech Stack: Identify and document the technology stack needed for the project.
B. Database Design: Design the database structure to support the application's data needs.
C. User Interface (UI) Design: Create wireframes and mockups for the application's UI.
D. System Architecture Design: Design the overall system architecture, including API and AI integration.

Development

A. Develop a Working Prototype: Build a functional prototype incorporating core features.
B. Test Prototype: Conduct initial testing to identify and resolve issues.
C. Review Prototype Design: Present the prototype to stakeholders for feedback.
D. Rework Requirements: Adjust requirements based on feedback from the prototype review.
E. Document Updated Design: Update design documents to reflect any changes made during development.

Final Report

A. Test Product: Perform final testing to ensure all project requirements are met.
B. Presentation Preparation: Prepare a presentation summarizing the project's objectives, process, and outcomes.
C. Poster Preparation: Create a poster to visually represent the project for presentations or exhibitions.
D. Final Report Submission: Submit the final report to both D2L and the project owner, including all relevant documentation and code.

Milestone Events (Prototypes, Draft Reports, Code Reviews, etc)

Requirements Finalization - By August 31, 2024

- Complete the requirements gathering process, review with stakeholders, and obtain sign-off.

System Design Completion - By September 14, 2024

- Finalize the system architecture, UI design, and database design.

Prototype Development and Review - By October 19, 2024

- Develop the working prototype and conduct initial testing. Review the prototype with stakeholders to gather feedback.

Final Testing and Analysis - By November 23, 2024

- Complete all testing phases, including performance, security, and cost assessments.

Final Report Submission - By November 30, 2024

- Submit the final report, including all documentation, source code, and project deliverables.

Meeting Schedule Date/Time

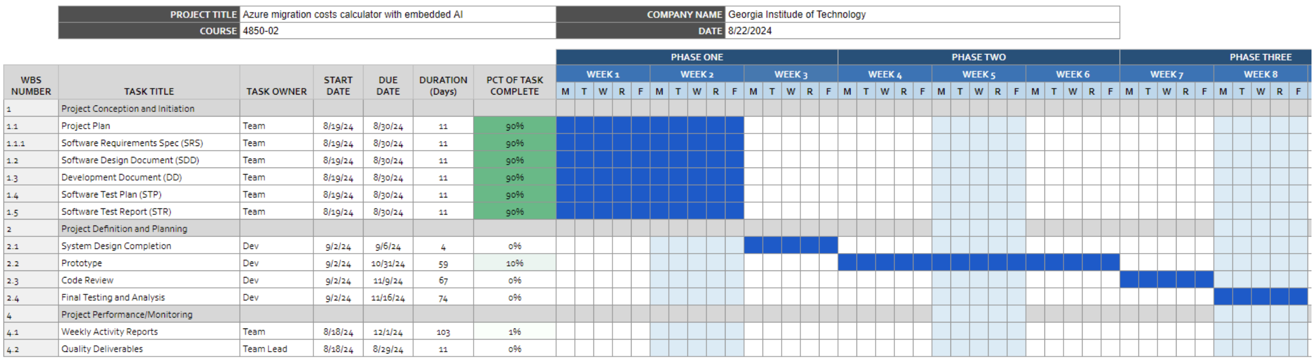
Professor Bi-Weekly meeting:	Thursdays at 7:15 PM EST.
Team Member Weekly meeting:	Friday at 1:30 PM EST.

Collaboration and Communication Plan

The project team will utilize MS Teams for formal communication, including bi-weekly status meetings. These meetings will serve as a platform for project updates, milestone tracking, and issue resolution. For day-to-day collaboration, the team will use Discord, which will host channels for general announcements, technical discussions, and resource sharing. Team members are expected to respond to messages within agreed-upon timeframes and keep project-related conversations in appropriate Discord channels for easy reference.

Document sharing will be managed through Google Docs with links in the appropriate Discord channels. For urgent matters, team members should use the @everyone tag in relevant Discord channels, with further escalation to the project manager via MS Teams if needed. Critical issues requiring immediate attention will be addressed through a predetermined method such as phone calls or SMS.

Project Schedule and Task Planning



GANTT CHART

PROJECT TITLE	Azure migration costs calculator with embedded AI	COMPANY NAME	Georgia Institude of Technology
COURSE	4850-02	DATE	8/22/2024

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION (Days)	PCT OF TASK COMPLETE	PHASE ONE			PHASE TWO			PHASE THREE	
							WEEK 1 M T W R F	WEEK 2 M T W R F	WEEK 3 M T W R F	WEEK 4 M T W R F	WEEK 5 M T W R F	WEEK 6 M T W R F	WEEK 7 M T W R F	WEEK 8 M T W R F
1	Project Conception and Initiation													
1.1	Project Plan	Team	8/19/24	8/30/24	11	90%	■■■■■	■■■■■						
1.1.1	Software Requirements Spec (SRS)	Team	8/19/24	8/30/24	11	90%	■■■■■	■■■■■						
1.2	Software Design Document (SDD)	Team	8/19/24	8/30/24	11	90%	■■■■■	■■■■■						
1.3	Development Document (DD)	Team	8/19/24	8/30/24	11	90%	■■■■■	■■■■■						
1.4	Software Test Plan (STP)	Team	8/19/24	8/30/24	11	90%	■■■■■	■■■■■						
1.5	Software Test Report (STR)	Team	8/19/24	8/30/24	11	90%	■■■■■	■■■■■						
2	Project Definition and Planning													
2.1	System Design Completion	Dev	9/2/24	9/6/24	4	0%			■■■■■					
2.2	Prototype	Dev	9/2/24	10/31/24	59	10%				■■■■■	■■■■■	■■■■■		
2.3	Code Review	Dev	9/2/24	11/9/24	67	0%							■■■■■	
2.4	Final Testing and Analysis	Dev	9/2/24	11/16/24	74	0%								■■■■■
4	Project Performance/Monitoring													
4.1	Weekly Activity Reports	Team	8/18/24	12/1/24	103	1%								
4.2	Quality Deliverables	Team Lead	8/18/24	8/29/24	11	0%								

Version Control Plan

The project will utilize Git as the version control system, hosted on GitHub. Our repository structure includes a main branch for stable, production-ready code, a staging branch for the latest prototype, development branches named after developers for feature integration, and bug fix branches for addressing issues.

Commits will be made regularly with descriptive messages, and changes will be integrated through pull requests (PRs) after review. Branch naming will be clear and descriptive, with prompt cleanup of unnecessary branches. Releases will be tagged using semantic versioning (e.g., V1.0.0). For continuous integration and deployment (CI/CD), we'll employ GitHub Actions or Jenkins to automate testing of new code and deployment after successful tests.