

Project Proposal Template

EE 547: Applied and Cloud Computing for Electrical Engineers

{**Project Title:**} [Your Project Title Here]

{**Team Members:**} [Name 1, Name 2, Name 3]

Introduction

Introduce your application and the problem it addresses. What does your system do? Who will use it and why? What functionality will it provide?

Be specific about what makes this problem appropriate for a cloud computing project. Why does it require multiple integrated components, asynchronous processing, and deployment to AWS?

System Architecture

Component Overview

List your major system components. For each component, what is its role? Which components handle: - User requests and API endpoints - Asynchronous background processing - Data storage - ML functionality - External service integration

How do these components communicate with each other?

Architecture Diagram

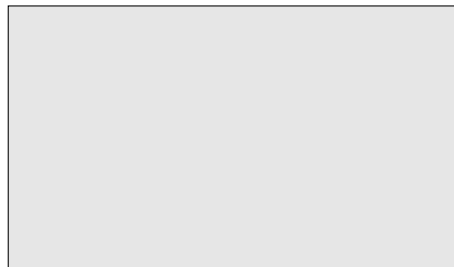


Figure 1: System Architecture Diagram - Replace with your architecture showing components, communication paths, data flows, and AWS services

Include a diagram showing: - Major components as labeled boxes - Communication paths (REST API calls, message queues, etc.) - External services or APIs - Databases and storage - Where users interact with your system

Technology Stack

What technologies will you use?

Languages: What programming languages for backend and frontend?

Databases and Storage: What database system (PostgreSQL, DynamoDB, etc.)? Why is this appropriate for your data? Will you use S3 for file storage?

AWS Services: What AWS services for compute, storage, messaging, etc.? Why these choices?

Frameworks: What frameworks for your API, frontend, ML, etc.?

Communication: How will components communicate? REST APIs? SQS queues? Other patterns?

Technical Details

REST API

What major API endpoints will you provide? For key endpoints, describe: - What they do - Request and response data - Authentication approach (JWT, sessions, API keys)

How will you handle errors and return appropriate status codes?

Asynchronous Processing

What operations must run asynchronously? Why can't they be synchronous?

What AWS services will coordinate async work (SQS, Lambda, workers on EC2)?

How will you communicate task status to users? How will you handle failures and retries?

Machine Learning Component

What ML technique will you use (classification, embeddings, recommendation, etc.)? What does it predict or generate?

Will you use a pre-trained model or train your own? What framework?

Where does inference happen in your architecture (synchronous API, async workers, batch processing)?

How does ML serve your application's purpose?

Data Persistence

What data needs to be stored (users, application state, ML outputs, files, etc.)?

What storage systems will you use and why?

Describe your data model at a high level. What are the main entities and their relationships?

How will components access stored data?

User Interface

What technology for your frontend?

What functionality will users access? What are the key user workflows?

How does authentication work? How do users track progress of async operations?

Data Sources and External Services

What external APIs, datasets, or services does your application depend on?

What are rate limits or usage constraints? How will you handle failures or unavailability?

If your application generates its own data through user interaction, explain that process.

AWS Deployment

How will you deploy to AWS?

Compute: EC2 instances? Lambda functions? What sizes? How many?

Storage: RDS database (what engine/size)? DynamoDB? S3 buckets?

Networking: VPC? How will components communicate?

Secrets: How will you manage API keys, passwords, credentials?

Deployment process: Manual console, CLI scripts, or infrastructure-as-code?

Implementation Plan

Team Responsibilities

What will each team member primarily implement? Ensure each member has substantial technical contributions.

Milestones

By Revised Proposal (09 Nov): What will you have validated or prototyped?

By Status Report (30 Nov): What should be deployed and working? What integration complete?

By Demo (11 Dec): System fully operational.

Include intermediate checkpoints that demonstrate progress.

Minimum Viable System

If you encounter delays, what minimum functionality still meets project requirements?

What can be simplified (simpler ML model, fewer features, reduced scale) while maintaining core architecture (multiple components, async processing, API, database, ML, AWS deployment)?

Technical Challenges

What specific technical challenges do you anticipate? For each major challenge, how will you address it?

Consider: external API reliability, ML training requirements, integration complexity, AWS costs, unfamiliar technologies, performance requirements.

Expected Outcomes

What will your completed system demonstrate? Describe the end-to-end user experience.

How does your project integrate course concepts (async processing, REST API, database design, ML integration, AWS deployment)?

What makes your project technically interesting or challenging?