**🚀 Group Project Instructions: Highly Durable N-Tier Web Application**

**Project Title:**  
**"Mission: (Highly) Available!"**

**Start Date:** April 22, 2025  
**End Date:** May 20, 2025  
**Team Size:** 2–4 members

**🎯 Project Objective**

Your team is tasked with designing, deploying, and testing a **highly durable N-Tier application** on AWS. The application should have the following characteristics:

* A **simple web front end** (static or dynamic—your choice)
* A **MySQL Amazon RDS backend**
* **CloudFormation (IaC)** to deploy resources
* **High durability & availability** with failover to a **Disaster Recovery (DR) region**

**🧩 Architecture Requirements**

**1. Web Tier**

* Static web page hosted on EC2, S3 (static), or Load Balanced EC2
* Must connect to backend via internal endpoint or API – (Optional)

**2. Application Tier *(optional)***

* Lambda function or EC2 instance(s) running app logic (can be simple PHP, Node.js, Python Flask, etc.)

**3. Data Tier**

* MySQL RDS instance
* Set to Multi-AZ deployment for high availability in the primary region

**4. Disaster Recovery (Automatic Failover with Cross-Region Replication)**

* Deploy a disaster recovery stack in a secondary AWS region
* Implement cross-region RDS read replica of your primary MySQL database
* Design your DNS failover using Amazon Route 53 with health checks to detect failure and redirect traffic to the DR region
* Web and App tiers must also be deployed and ready to serve traffic in the DR region
* Simulate a primary region failure and demonstrate automatic failover with data integrity preserved via replication
* Document your failover process and recovery point objective (RPO) and recovery time objective (RTO) assumptions

**📅 Project Milestones**

| **Milestone** | **Due Date** | **Deliverables** |
| --- | --- | --- |
| 🗂️ Sprint Plan | April 24, 2025 | Timeline, tasks, assignments, and expected sprint outcomes (PDF or DOCX) |
| 🏗️ CloudFormation Template (IaC) | April 29, 2025 | Deploys primary region N-Tier architecture (YAML or JSON with README.md) |
| 🌐 Primary Region Deployment | May 6, 2025 | Functional web app with RDS backend in primary region (demo or screenshot) |
| 🌍 DR Region Deployment & Failover | May 13, 2025 | Deployed DR stack with DNS/manual failover demonstration |
| 📦 Final Submission & Presentation | May 20, 2025 | Final docs, GitHub link, architecture diagram, and a 5-minute presentation |

**🔧 Tools and Technologies**

* AWS Services: EC2, S3, RDS (MySQL), CloudFormation, Route 53, Lambda (if used), IAM, CloudWatch
* IaC: AWS CloudFormation (YAML/JSON)
* Version Control: GitHub or GitLab repo
* Project Management: Trello, Jira, or a shared Google Doc (your choice)

**📚 Documentation & Deliverables**

Each group must provide the following by the end of the project:

* 📝 Sprint Plan (with roles, dates, user stories, backlog items)
* 📜 CloudFormation templates with inline comments
* 🧱 Architecture Diagram (Lucidchart, draw.io, or any visual tool)
* 🌐 URL or video/screenshots showing the app working in both regions
* 📘 A brief summary report covering:
  + Team roles and responsibilities
  + Challenges faced & how you overcame them
  + Improvements you would make if given more time

**🧠 Tips for Success**

* Use **Multi-AZ RDS** for durability in your primary region.
* Implement **Route 53 health checks** if you want a slick DR failover demo.
* Use **CloudFormation exports/imports** to link stacks (i.e., app stack imports DB info).
* Communicate regularly as a team.