CSCI 5411 Mid-Term Project
Summer 2025
AI/GenAI on Cloud

Overview

This project is developed for students to architect and implement an AI/GenAI solution on AWS Cloud, demonstrating their understanding of cloud architecture concepts with a specific focus on artificial intelligence services. Students will leverage AWS's AI/ML ecosystem to build a practical application that shows both their cloud architecture skills and their ability to work with AI models.

The project encompasses **three components**:

- A compulsory AI/GenAI application design and implementation.
- A compulsory one-on-one meeting to demonstrating your projects.
- An optional and introductory AWS DeepRacer component (3 bonus point for this term project) designed to familiarize students with fundamental AI concepts through hands-on experience.

Note

Your instructor covered general AI/GenAI concepts, and application architectures in CSCI4145/5409 in the winter term 2025. If you did not take that course, you may need to review the materials independently. All related lecture slides and recordings are available on Brightspace. Please go through them at your own pace, and don’t hesitate to reach out with any questions. If you have learned AI/GenAI from other sources, you may skip the slides and recordings.

If you previously developed an AI/GenAI term project in CSCI4145/5409, you are required to create a completely different AI/GenAI project for this course. This is a great opportunity to explore new areas of AI that you haven’t worked with before.

Objective

The project is designed to understand your demonstration in architecting cloud-based AI/ML solutions using AWS services and for you to gain experience with model development, deployment, and management in a cloud environment. The project will focus on security, cost and debugging related practices that you will apply to your application. It is designed for you to understand technical and operational considerations specific to AI/GenAI applications in cloud environments. And for you to experience the complete AI development lifecycle from data exploration to model deployment.

Scope

1. You **DO NOT** need to write Infrastructure as Code (IaC) for this project. However, if you wish to do so, you may, but no extra points will be awarded for these actions.
2. The introductory AWS DeepRacer component will serve as a primer on AI concepts and will be completed early in the project timeline to establish foundational understanding before tackling the main implementation.
3. Students can utilize pre-existing open-source AI/ML models or frameworks as a starting point. If using existing models, these should be from reputable sources with sufficient community validation.
4. You are expected to use AWS Academy Learner Lab resources allocated for this course. While the $50 AWS Academy credit should be sufficient, students must carefully monitor and manage their resource usage throughout the project.

5. If you opt to employ your personal AWS account, that's perfectly acceptable; however, I urge you to exercise utmost caution regarding your incurred costs. Please note that AWS Academy Learner Lab have a lot of restrictions on their services. You will have to understand the restrictions before architecting your application. You can find all supported services on the right side of the interface after you launch the learner lab.

Project Distribution Overview (component weights in the project)

Time management is essential in a professional environment. However, many students tend to start their projects at the last minute. To address this, we've implemented milestones, like industry sprints, to help pace the project. Students are encouraged to start early, especially given the compressed term.

Below is the overall weight distribution of all the components:

Weight Distribution			
Optional DeepRacer Model	**Project Report**		**One-on-One Meeting**
3-points bonus	**70%**		**30%**
	Milestone 1	Milestone 2	
	20%	80%	

Note: The 3-point bonus of the DeepRacer model won't make the overall marks of the project over 100.

Optional: AWS DeepRacer (3-point Bonus)

- **Due Date:** 11:59pm, May 11
- **Deliverable**: Trained DeepRacer model and performance demonstration. The details of this component are at the end of this doc.
- **Note**: This component is optional but can provide 3 points toward your final grade of this project.

Milestone 1: Exploration, Architecture and Design (20% of the project report)

- **Due Date:** 11:59pm, May 18
- **Deliverables**:
  - AI/GenAI application domain selection and justification. Problem statement and business case for selected AI/GenAI application.
  - Detailed functional requirements specification.
  - Non-functional requirements analysis. Analysis of security and scalability must be included. You should also include all relevant non-functional requirements necessary for your project.
  - Initial architecture design with diagram. You must provide a diagram for your architecture. The design and architecture can be slightly improved and modified as you develop your project in the second milestone.
  - Besides architecture design, you also need to provide a data sequence diagram (also called sequence diagram or UML data sequence diagram) to show how data and messages flow between different components, objects, or systems over time.

    Note: You can use draw.io to draw the application architecture and data sequence diagram. But you are free to use other tools. You must explain your architecture and data sequence diagram. You are welcome to include any additional diagrams that help clarify and explain your project.

- AWS services and tech stack. Briefly describe the programming languages, AWS services and tools, as well as any additional technologies you plan to use for implementing your project. You don't need to explain everything in detail at this stage, as you will provide a more comprehensive explanation in the second milestone.
- Identification of potential architectural challenges.
- Initial cost estimation for overall implementation.

Milestone 2: Implementation and Documentation (80% of project report)

- **Due Date:** 11:59pm, June 1
- **Deliverables**:
  - Final report on top of the report of milestone 1.
  - Improved architectural diagram and data sequence diagram. There shouldn't be huge changes to your architecture design and data sequence diagram. You keep your original diagrams if you do not need to change them.
  - Detailed explanation of the implementation of the designed project:
    - Service configuration details and screenshots of proper AWS resource configuration and management
    - Model selection/training/enhancement/improvement/integration process
    - Implementation of appropriate monitoring and logging solutions
    - Demonstration with screenshots showing the model (developed or imported), its integration, and how it functions within the system
    - Testing methodology and results
    - Security measures at all layers
    - Cost analysis and optimization strategies
    - Lessons learned and future improvements

Major tasks for the term project:

1. Select the project domain and based on the domain you either need to develop your own model or use an existing one.
2. If you are going to use an existing model, you need to deploy this model in either AWS SageMaker (SageMaker is available on Learner's lab with some restrictions) or EC2 with appropriate measures.
   NOTE: Calling an endpoint (e.g. OpenAI, Gemini etc.) will not be considered for this project. You can integrate it to enhance your project, but you cannot develop your project around that LLM's endpoint. For example, simply designing an interface to collect user prompts, calling ChatGPT-like APIs, and displaying responses—without incorporating the required techniques—will not be considered a valid project for this course. The required techniques are explained in the next point.
3. You need to explore and implement domain such as RAG, prompt engineering, in-context learning, vector databases, fine-tuning etc. for your project to be considered as a new AI project.
4. Design the cloud architecture to host your AI application on AWS. When designing the architecture for the chosen application, it is imperative to adhere to the principles and best practices for security, cost and troubleshooting/governance/monitoring.

5. While there are no restrictions on the services you choose to use, your architecture must include appropriate monitoring and governance tools. These should enable you to effectively trace and troubleshoot issues across your entire system and each individual service you incorporate.
6. AI projects can be expensive on the cloud, we require you to calculate the price for the project in the long run and apply cost optimization measures for the services and architecture as well.
7. Due to restrictions from Learner's Lab, we won't require you to apply all the best security measures, but you need to have the knowledge of what can be done for security measures and implement the security measures at architecture and service level as much as possible. You can explain the security measures that you are not able to implement in the learner lab.
8. Implement your planned architecture on AWS. You may also choose to integrate other cloud platforms such as GCP or Azure to enhance your project, but using a multi-cloud environment is optional.

AWS Managed AI Services

AWS offers managed AI services like ChatGPT, Gemini, and others for various tasks. However, for this term project, you are not allowed to use AWS managed AI services as the core of your AI solution. You must either develop your own model or use an existing one via SageMaker or an EC2 instance. That said, you are welcome to use AWS managed AI services to enhance or complement your overall project, as they can significantly contribute to building a well-rounded and effective final product.

The below services are tagged as AWS Managed AI Services:

- Text and Document - Amazon Comprehend, Amazon Translate, Amazon Textract
- Vision - Amazon Rekognition
- Search - Amazon Kendra
- Chatbots - Amazon Lex
- Speech - Amazon Polly, Amazon Transcribe
- Recommendation - Amazon Personalize
- Miscellaneous - Amazon DeepRacer

AWS Bedrock

AWS Bedrock provides access to multiple different foundation models in a semi-managed environment, allowing you to leverage pre-trained models for your specific use cases. While this offers significant capabilities, please note that AWS Academy Learner Lab environments do not support Bedrock access.

If you choose to use your personal AWS account to incorporate Bedrock into your project, be aware that your evaluation criteria will differ. Our assessment will focus heavily on your foundational model selection process, including comparative evaluation of different models, implementation of GuardRails for responsible AI use, any fine-tuning techniques applied, configuration of Bedrock agents, and cost management strategies.

Cost consideration is particularly important with Bedrock due to the potentially high expenses associated with Provisional Throughput. You should demonstrate thorough understanding of pricing implications and implement appropriate cost optimization measures. While using Bedrock is permitted, the comprehensive management requirements and associated costs make this approach more demanding from an evaluation perspective compared to other implementation options.

Amazon SageMaker

SageMaker is available on the Learner's lab with some restrictions. You can read learner lab documentation for a better understanding of what you can do on SageMaker and how you can configure SageMaker before you start your project.
Only Open-Source models are available in the SageMaker JumpStart. You can deploy any open-source models in the following supported SageMaker instance types: ml.t3.medium, ml.t3.large, ml.t3.xlarge, ml.m5.large, ml.m5.xlarge, ml.c5.large, ml.c5.xlarge only. There might be many open-source models that you can use but the model might be too big for the above provided instance types.

Make sure to understand and estimate the cost while running the notebooks on SageMaker. Make sure that you setup the SageMaker Domain as per the documentation from learner's lab. If you are using SageMaker I would highly recommend to explore Augmented AI, Data Wrangler, Hyperparameter tuning jobs, Model Dashboards and Model Cards. There are lots of options that SageMaker provides you to ease out your ML pipeline, do try to explore all of them.
SageMaker Canvas is available in learner's lab, but it is highly restrictive, and you might not be able to complete your task using Canvas. Hence, I would recommend explore the service first and understand the restrictions.

Deliverables

Project Report (70% of total project)

A detailed report outlining the choices you made and justifying them. Justify your choice of services. Explain why the chosen service was the best fit for your application, considering factors like cost, security, and scalability.

This report should also explain how your architecture adheres architecting principles and best practices. Inclusion of an architecture and sequence diagram is expected in this report.

The report must present the hosted application's public URL or other accessible evidence of the running application on AWS. Marker might open the same URL during the meeting.

Evaluation and Rubrics

- You will be evaluated based on the understanding of the services you have utilized.
- Evaluation will consider the cost, security and other architectural related decisions for your project.
- You will be evaluated on the choice and understanding of your model and the knowledge on the AI domain that you have used.
- You will be penalized for presenting any misleading information during the meeting or in the report, and thus you need to make sure that you only represent your actual architecture and other related practices.
- Any irrelevant services that you have utilized in your project will be considered during evaluation and based on that you will be penalized.
- You will be evaluated based on the governance and monitoring services that you have used and the way you have utilized them.

Note: If multiple students submit very similar or identical architectures, the instructor will schedule individual meetings to investigate potential academic misconduct.

One-on-One meeting evaluation (30% of total project)

Meetings will be from 30th May to 1st June.

Note 1: At the beginning of the video, please ensure that your student ID card is visible in front of the camera and keep the camera active throughout the entire meeting. Failure to comply with this will result in a score of **ZERO**.

Note 2: The meeting is approximately 15-20 minutes long. Students are expected to use around 7-8 minutes to present their design and discuss the architecture's implementation. The markers will ask questions during or after the presentation. Evaluation is based on the quality of the presentation and the accuracy and thoroughness of the responses to the questions.

Note 3: If a student struggles to answer questions adequately or their responses fail to clarify their design and implementation, the markers will report this to the instructor for further investigation. Suspected cheating can lead to severe consequences, including a grade of **ZERO**.

1) Presentation (**5 marks**)
    a. The presentation is thoughtfully planned, follows a logical structure, and communicates the information clearly. (1-3 marks)
    b. The presentation seamlessly corresponds with the content of the report and milestones. (2 marks)

    Justification (the following will result in either 1 or 0 mark):

    1) The presentation partially correlates with the content of the report. (1 mark)
    2) The presentation is unrelated to the report. (0 mark)
2) Question Answering (**10 marks**)
    a. All responses are clear, accurate, and consistent with the presentation and the report. (10 marks)
    b. The responses lack clarity, contain inaccuracies, or contradict the content of the presentation and report. (1-9 marks)
    c. The student is unable to answer all questions; most responses are incorrect, or they are unrelated to the content of the presentation and report. (0 mark)

Submission

Report submissions will be made on Brightspace by the due date. One-on-One meetings will be scheduled by the marker.

FAQ:

1. How long should be the report?

There isn't a required length of the report. The length of your report will be reasonable if you try your best to provide a clear explanation for your choices and decisions.

2. Can I submit my milestones and report multiple times?

Yes, you can. If you have submitted your assignment, but later found that you need to change it, you can just upload a new file. Only the most recent submission will be kept on Brightspace.

3. Will I lose points if I do not include diagrams?

Architecture and sequence diagrams are expected deliverables of this assignment, and you will lose points if you do not submit these deliverables.

CSCI 5411 AWS DeepRacer – Summer 2025

Overview

Build your model, evaluate its performance on a virtual track, and then compete in the AWS DeepRacer Virtual Circuit.

The fastest way to learn basics of ML

AWS DeepRacer allows developers of any skill level to get started with machine learning with hands-on tutorials and guidance on building reinforcement learning models. Reinforcement learning is a branch of machine learning, ideal for a variety of practical business problems from robotics automation, to finance, to game optimization, and autonomous vehicles.

Quickly evaluate models in the 3D racing simulator

AWS DeepRacer has an integrated simulation environment hosted on the AWS Cloud for experimentation and optimization of your autonomous racing models, built with reinforcement learning. Train your models and get ready to race in the Virtual Circuit.

Assignment

This mini assignment is based on AWS DeepRacer. The main objective of this mini assignment is for you to learn how the model is developed, how it is evaluated and few other terminologies related to Machine Learning (Artificial Intelligence). AWS DeepRacer is available on Learner’s lab.

Most of the part of this assignment does not require any kind of coding. But there might be some level of coding involved where you need to write the logic for the reward function at the end while developing the model.

You can take a small reinforcement learning crash course that is provided by AWS on the AWS DeepRacer Console after that you can begin your model development journey.

We are going to host a small race after the deadline, the race details will be shared via the Teams Channel. Race will be for fun and evaluating your model.

Model Development (Model Training)

Environment simulation

You can choose any track of your choice from console for training, but make sure to choose the track direction as clockwise while training the model.

Note: You can train your model on any track, but the race is going to be hosted on “European Seaside Circuit - Buildings track “.

Race type and training algorithm

You can choose **Head-to-head racing** which is vehicle races against other moving vehicles on a two-lane track during the training phase. Now comes the configuration part where you can choose your preferred type of configuration i.e. Training algorithm and hyperparameters.

NOTE: All the configurations are completely on you and your decisions that you take.

Define action space

This choice of choosing action space is completely based on your preference and your decision while developing a model.

Reward Function

As a part of evaluation, your reward function will be assessed and understood very well, so you need to write the reward function based on your decisions while developing the model.

NOTE: You can evaluate your model, and you must evaluate your model after training to find out how the model is performing overall.

Deliverables and Evaluation

You need to submit 1 page justification on Brightspace regarding all the decisions that you took while training the model. You need to explain your reward function that you write while training phase in this submission.

Through the teams channel we will provide you the information to submit your model for the Virtual race that will be held after the deadline. The performance of the model will also be observed in the race and will be used for evaluation.

Pricing

With AWS DeepRacer, you are charged for storage, and training and evaluation jobs. Model training and evaluation is charged at a flat fee of $3.5 per hour and model storage is $0.023 per Gigabyte per month.

The AWS DeepRacer Free Tier provides 10 free hours to train or evaluate models and 5GB of free storage during your first 30 days to create, train, evaluate, and submit your model to the Virtual Circuit.