GenAI-Powered Systems: A Generative AI Campus Assistant and a Voice-Controlled Autonomous Robot

ELECOMP Capstone Design Project 2026-2027

Sponsoring Company:

Amazon Web Services (AWS)

410 Terry Avenue North Seattle, WA 98109
Website: https://aws.amazon.com
AWS Public Sector: https://aws.amazon.com/government-education/

Company Overview:

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies and universities—use AWS to lower costs, become more agile, and innovate faster. AWS provides a deep portfolio of services spanning compute, storage, databases, analytics, machine learning and generative AI, Internet of Things (IoT), robotics, and edge computing.

AWS is deeply committed to education and workforce development. Through this ELECOMP Capstone sponsorship, AWS aims to give URI's senior Electrical and Computer Engineering students hands-on experience building real-world systems powered by generative AI—mirroring the architectures that power modern intelligent products.

Technical Director(s):

Fang Cong
Sr. Solutions Architect, AWS WWPS SLG/EDU East
fcong@amazon.com

Brandon Artz
System Dev Engineer, RDPI Hardware Solutions, Amazon
branartz@amazon.com

Consulting Technical Director(s):

Hui Lin
Associate Professor
Electrical and Computer Engineering Department, University of Rhode Island
huilin@uri.edu

Project Motivation:

Generative AI is reshaping how software is built and how people interact with machines, yet most engineering graduates leave school having only *used* AI tools—not having *built* systems on top of them. This sponsorship is designed to close that gap. It gives URI's ELECOMP seniors direct, hands-on experience architecting production-grade generative-AI systems on a leading cloud platform, developing precisely the skills employers are urgently hiring for.

This sponsorship proposes **two projects** to be delivered by **a single combined student team of 4-5 ELE/CPE seniors**. Both projects share a common theme—**GenAI-powered systems built on Amazon Bedrock**—allowing the team to build reusable cloud and generative-AI skills (AWS account setup, identity, serverless integration, Amazon Bedrock, RAG, voice pipelines) once and apply them across both efforts. The two projects are deliberately chosen to span the spectrum of applied GenAI:

- **Real-world relevance and workforce readiness.** Retrieval-Augmented Generation (Project A) and LLM-driven task planning (Project B) are two of the most in-demand patterns in industry today. Students graduate fluent in the exact architectures powering modern AI products.

- **Full-stack, interdisciplinary engineering.** Project A exercises cloud-native software and responsible-AI practices; Project B adds embedded hardware, sensors, power, and real-time control. Together they let ELE and CPE students each contribute their strengths on one cohesive team.
- **Lasting institutional value.** The Campus Assistant is an immediately useful tool the College of Engineering can continue to operate, and the robot is a compelling, demonstrable showcase for the Spring Summit, recruiting, and future capstone recruitment.
- **A reusable foundation.** Because both projects build on the same Amazon Bedrock core, the team invests once in cloud/AI fundamentals and applies them twice—maximizing learning and de-risking delivery within the academic year.

Publication & open-source opportunity: Students will have the opportunity to co-author a technical blog post with their AWS mentors documenting the project and its results, with the goal of publishing it on the AWS website (e.g., the AWS Public Sector or Architecture blog). The resulting solutions can also be open-sourced so that other universities and organizations can adopt and benefit from the students' work—amplifying the impact of the capstone beyond URI.

PROJECT A: Generative AI Campus Assistant

Anticipated Best Outcome:

A production-quality conversational assistant for URI's College of Engineering that answers student and faculty questions about courses, program requirements, lab schedules, department policies, and campus events—grounding every response in authoritative university documents using Retrieval Augmented Generation (RAG) and citing its sources. The best outcome is a deployed, authenticated web application that achieves at least 80% answer accuracy on a curated evaluation set, responds in under three seconds, and is ready for the College of Engineering to operate.

Project Details:

The team will design and build an AI-powered conversational assistant for URI's College of Engineering. Students and faculty frequently need answers scattered across dozens of university documents—course catalogs, handbooks, lab safety policies, advising guides, and event calendars. This assistant serves as a single, intelligent point of access to that institutional knowledge, while avoiding the hallucination problems of ungrounded chatbots by grounding responses in source documents and citing them.

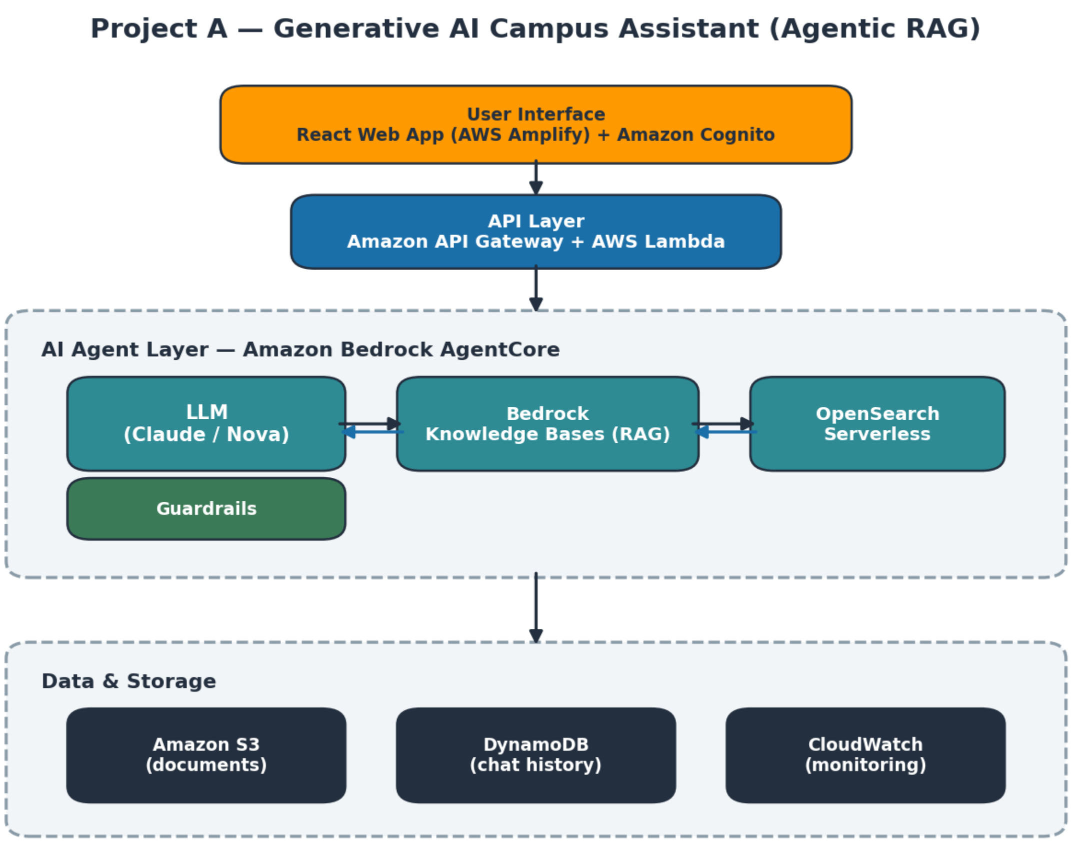

Figure A-1: Generative AI Campus Assistant System Architecture.

The project is organized around the following outcomes:

ABO A-1: Document Ingestion and Knowledge Base- Collect and curate URI College of Engineering documents (course catalogs, handbooks, lab schedules, policies) - Chunk, embed, and index documents into **Amazon Bedrock Knowledge Bases** with a vector store (**Amazon OpenSearch Serverless**) - Store source documents in **Amazon S3**; implement an automated refresh pipeline so new/updated documents are re-indexed on a schedule - Target: at least 50 documents indexed with a validated chunking strategy

ABO A-2: Conversational AI Agent with Grounded, Cited Responses- Build an agentic assistant using **Amazon Bedrock AgentCore** with retrieval actions over the knowledge base - Include a source citation in every response, linking back to the originating document - Apply **Amazon Bedrock Guardrails** to block off-topic or inappropriate queries and reduce hallucination - Support multi-turn conversation with context retention; target at least 80% accuracy on a 100-question evaluation set

ABO A-3: Web Interface and Deployment- Build a web-based chat interface (**React**) hosted on **AWS Amplify**; authenticate users via **Amazon Cognito** (URI email login) - Store conversation history in **Amazon DynamoDB**; expose backend logic via **AWS Lambda** and **Amazon API Gateway**- Provide a feedback mechanism (thumbs up/down); target sub-3-second response time and mobile-responsive design

ABO A-4: Evaluation Framework and Continuous Improvement- Build an automated evaluation harness that scores answer accuracy and citation correctness against a curated question set. Establish a continuous-improvement loop feeding evaluation results back into prompt and retrieval tuning

ABO A-5 (Stretch): Voice Interface- Add speech-to-text input and text-to-speech output (**Amazon Transcribe** + **Amazon Polly**) for accessibility—reusing the voice pipeline shared with Project B

PROJECT B: Voice-Controlled Autonomous Robot with Cloud Intelligence

Anticipated Best Outcome:

A working autonomous mobile robot that accepts spoken natural-language commands, interprets and plans multi-step tasks using a cloud-hosted LLM, and executes physical actions—navigating while avoiding obstacles, capturing and interpreting images, and reporting results back verbally. The best outcome is a robot that reliably executes at least five distinct command types end-to-end and is demonstrable live at the Spring Summit, backed by a real-time mission-control dashboard.

Project Details:

The team will design and build an autonomous mobile robot controlled through natural-language voice commands. A user speaks a high-level instruction (e.g., "go to the window and tell me if the blinds are open"); the robot streams the audio to the cloud, where Amazon Transcribe converts speech to text and Amazon Bedrock (an LLM) interprets the intent and decomposes it into an executable sequence of actions. The robot executes those actions—navigating while avoiding obstacles, capturing images, analyzing its surroundings—and reports back verbally via Amazon Polly. A web-based mission control dashboard provides live status, a camera feed, a map of the robot's path, and command history.

This project demonstrates a modern **edge-cloud hybrid architecture**: the resource-constrained robot handles real-time control and safety locally, while offloading heavy AI reasoning (language understanding, task planning, visual scene interpretation) to the cloud.

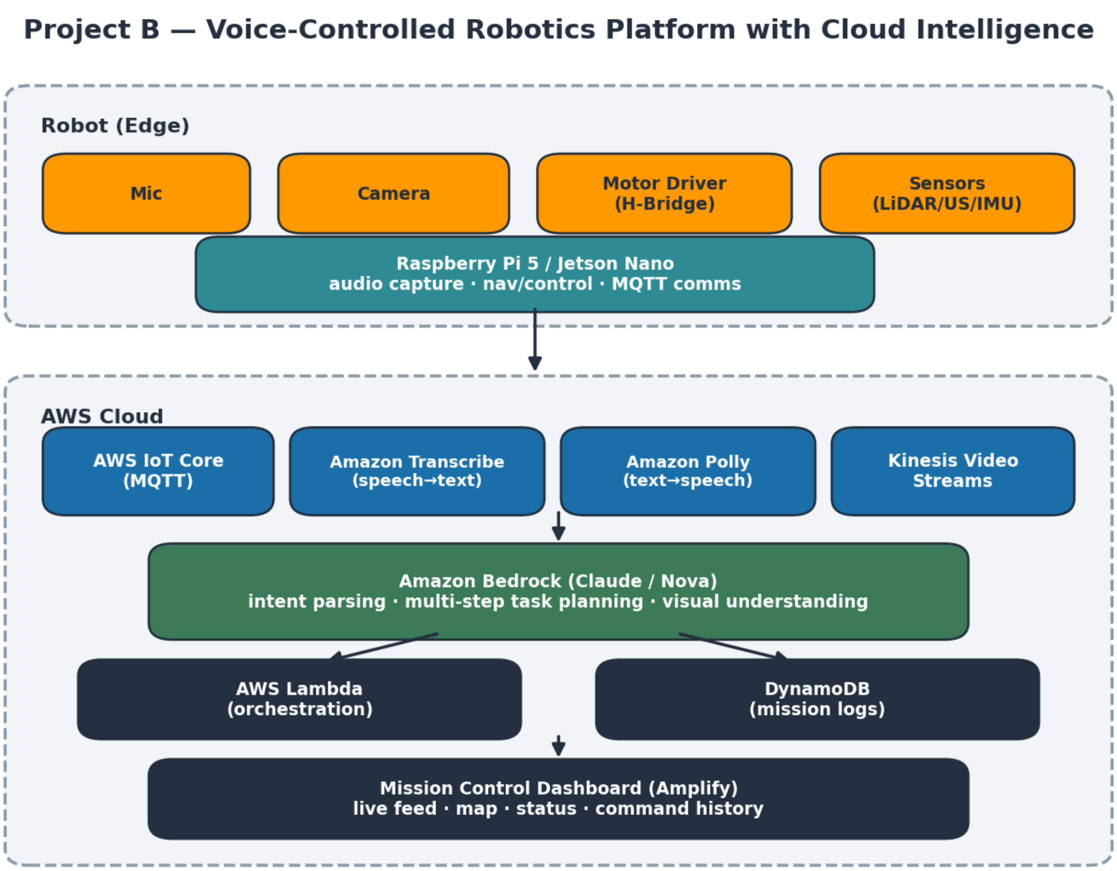

Figure B-1: Voice-Controlled Robot System Architecture.

The project is organized around the following outcomes:

ABO B-1: Mobile Robot Platform with Autonomous Navigation

Hardware / Electrical Tasks:- Design and assemble the robot chassis with differential drive (two DC motors + caster), including motor driver (H-bridge) circuitry - Integrate wheel encoders for odometry and an IMU for orientation sensing - Integrate obstacle-detection sensors (ultrasonic and/or LiDAR) - Design the power subsystem: LiPo battery pack, voltage regulation, and a power-monitoring circuit - (Optional) Integrate a servo-driven gripper/manipulator for pick-and-place tasks

Firmware / Software / Computer Tasks:- Program the on-board controller (Raspberry Pi 5 or NVIDIA Jetson Nano) for motor control and sensor acquisition - Implement obstacle avoidance and waypoint navigation using odometry; establish teleoperation over Wi-Fi as an initial milestone

ABO B-2: Voice Command Understanding and Task Planning (Cloud AI)- Capture microphone audio on the robot and stream it to the cloud; integrate **Amazon Transcribe** for speech-to-text - Integrate **Amazon Bedrock** (Claude or Amazon Nova) with a system prompt that decomposes a high-level command into an ordered sequence of executable robot actions (e.g., navigate_to → capture_image → analyze → report) - Integrate **Amazon Polly** for spoken responses; establish robot-to-cloud messaging via **AWS IoT Core** (MQTT); support at least five distinct command types end-to-end

ABO B-3: Visual Understanding of the Environment- Capture camera frames and send them to a Bedrock multimodal model for scene description and object detection - Enable the robot to answer environment questions (e.g., "how many chairs are in the room?", "is the light on?") and fuse visual results into the task-planning loop

ABO B-4: Mission Control Dashboard- Build a web dashboard hosted on **AWS Amplify** showing live robot status (battery, position, current task) - Stream the live camera feed via **Amazon Kinesis Video Streams**; display command history and a map of the robot's path - Store mission logs in **Amazon DynamoDB**; orchestrate cloud logic with **AWS Lambda**

ABO B-5 (Stretch): Multi-Step Autonomous Task Chaining- The LLM decomposes complex commands into multiple sequential sub-tasks executed autonomously (e.g., "check all three rooms and report back which ones are occupied")

Composition of Team:

This combined effort is delivered by a single team of **2 Electrical Engineers and 2 Computer Engineers**.

For Project B, preference will be given to seniors enrolled in the PCB Design Course, taught by Mike Smith.

Skills Required:

Students entering this project should have, or be prepared to develop, the following skills. All 3 Technical Directors will provide guidance and enablement on the cloud- and AI-specific areas.

- **Programming:** Proficiency in Python and JavaScript/TypeScript; familiarity with Git and collaborative development
- **Cloud & Generative AI:** Interest in (and willingness to learn) Amazon Bedrock, RAG, prompt engineering, and serverless architecture (Lambda, API Gateway, DynamoDB)
- **Web development:** React front-end development; REST API integration; basic authentication concepts
- **Embedded systems (Project B):** Microcontroller/single-board-computer programming (Raspberry Pi/Jetson), sensor interfacing, real-time control
- **Electrical/hardware (Project B):** Circuit design, motor driver and power-supply design, PCB layout, and instrumentation

Anticipated Best Outcome's Impact on Company's Business, and Economic Impact

The anticipated best outcomes create economic impact along several dimensions:

- **Talent pipeline.** The projects produce graduates fluent in Amazon Bedrock, RAG, serverless, and AI—skills in acute demand across AWS and its customers and partners. This strengthens the hiring pipeline for AWS and the broader cloud economy.
- **Reference architectures and reusable assets.** The Campus Assistant (RAG on Bedrock) and the robot's LLM task-planning stack are reusable reference implementations. Published as a blog and/or open-sourced, they lower the cost for other institutions and customers to adopt the same patterns, accelerating AWS service adoption.
- **Brand and community impact.** A co-authored AWS blog post and a demonstrable Spring Summit showcase raise AWS's visibility in higher education and provide proof points that can be reused across the education field. The open-source solutions extend the economic benefit to the wider community that adopts them.

Broader Implications of the Best Outcome on the Company's Industry:

Beyond AWS, the outcomes have implications for the cloud, AI, and education-technology industries:

- **Advancing responsible, grounded AI in higher education.** The Campus Assistant models a responsible pattern—retrieval-grounded answers with citations and guardrails—that directly addresses higher education's concerns about hallucination and trust in AI. A published, open-source blueprint helps set a practical standard other institutions can follow.
- **Democratizing applied generative AI.** By open-sourcing the solutions and documenting them in a public AWS blog, the work lowers the barrier for smaller universities and organizations—those without large AI teams—to deploy their own grounded assistants and LLM-driven automation, broadening participation in the AI economy.
- **Edge-cloud AI for robotics.** The robot demonstrates how cloud-hosted LLMs can augment resource-constrained edge devices for natural human-robot interaction—an architecture increasingly relevant across manufacturing, logistics, healthcare, and consumer robotics.
- **Workforce transformation.** Producing engineers who can *build* on generative AI (not merely use it) helps the industry close a critical skills gap, informing how engineering curricula and industry-academia partnerships evolve to prepare the next generation of AI-fluent engineers.

Appendix A: AI Use Policy

Allowed

Students may use AI tools for

- ✓ brainstorming support
- ✓ concept development
- ✓ literature and publicly available documentation review
- ✓ software development
- ✓ debugging
- ✓ data analysis
- ✓ test development
- ✓ non-proprietary paragraph editing
- ✓ non-proprietary single-slide formatting
- ✓ preparation of project deliverables

Not Allowed

- Note taking or transcription of meetings
- Using AI in violation of university policies, applicable laws, or course-specific rules identified by the capstone instruction team and project-specific rules identified by the technical directors.
- Entering proprietary, confidential, personally identifiable, security-sensitive, or otherwise restricted information into an AI system without authorization.
- Submitting AI-assisted work without understanding, reviewing, verifying, and appropriately disclosing its use.
- Submitting entire documents or presentations to an AI system for editing or formatting.
- Any unchecked items from the allowed list above.

Requirements

- If any use is allowed, AI tools may only be used to enhance, but not replace, students' technical reasoning, engineering judgment, and understanding. Students must be able to explain, justify, and defend all submitted designs, code, analyses, test results, documentation, and conclusions.
- During the first two weeks of the project, the student team shall work with the Technical

Directors to: identify any proprietary or confidential information that may be involved in the project; determine what information may or may not be shared with AI systems; identify the AI tools, models, accounts, and configurations approved for project use; and document any project specific restrictions or approval requirements.

- Students shall transparently document the uses of AI using the URI CYPHER AI Usage Marking guidelines (https://web.uri.edu/cypher/ai-marking/). A workshop is planned for September to familiarize students with these guidelines.
- Students retain full professional responsibility for all design decisions, analyses, implementations, testing, documentation, presentations, and final deliverables, regardless of whether AI tools contributed to the work.