

AWS Hackathon Tutorials and Guidelines

1. Set Up Your AWS Account & Apply Credits

- **Create or use an existing AWS account.** Creating a new AWS account is recommended such that you can get free credits to build your project on AWS.
- These credits are essential for using services like **Amazon Bedrock, AWS Lambda**, and others required for your project.

2. Build Agents for Amazon Bedrock Agentcore

The core of the project will be building a functional AI agent.

- **Follow the tutorials and developer guides** to learn how to launch, invoke, and manage your agent.
- These resources cover everything you need to know about using **Agents for Amazon Bedrock**.
- **Key Resource:**
 - AWS Agentcore: <https://aws.amazon.com/bedrock/agentcore/>
 - Developer Guide for building agents in AWS bedrock agentcore: <https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/what-is-bedrock-agentcore.html>
 - Agentcore tutorials and Samples: <https://github.com/aws-labs/amazon-bedrock-agentcore-samples/>
 - How to host agents: <https://github.com/aws-labs/amazon-bedrock-agentcore-samples/tree/main/01-tutorials/01-AgentCore-runtime/01-hosting-agent>
 - Agentcore toolkit to launch agents in Bedrock agentcore: <https://aws.github.io/bedrock-agentcore-starter-toolkit/>
 - How to invoke an agent : <https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/runtime-invoke-agent.html>

3. Submission Guidelines

What to Submit

- **Project Report:** A comprehensive document (PDF or docx format preferred) detailing the project.
- **GitHub Repository Link:** The complete source code for your project must be available in a public GitHub repository.
- **Live Demo Link:** If you've hosted your application, provide the direct URL.
- **Short Video Demo Link:** A 3-5 minute video showcasing your project's features and explaining how it works. You can host this on YouTube (as an unlisted video).

Report Content Guidelines

1. **Team Name & Members:** List your team's name and all participating members.
2. **Project Title:** A creative and descriptive name for your project.
3. **Problem Statement:** Briefly describe the problem or challenge your project aims to solve (1-2 paragraphs).
4. **Solution Overview:** Explain your solution. How does your project address the problem? What does it do?
5. **Technology Stack:** List the key technologies, frameworks, APIs, and AWS services (e.g., Amazon Bedrock Agents, AWS Lambda, Python, React) you used.
6. **Key Project Links:** This is a critical section. Consolidate all essential links here for easy access:
 - a. **GitHub Repository:** The direct link to your code.
 - b. **Live Application URL:** The link to your deployed website or application.
 - c. **Setup & Running Instructions:** A direct link to the README.md file in your repository. This file **must** contain clear, step-by-step instructions on how to set up and run the project locally.
 - d. **Video Demo URL:** The link to your presentation or demo video.
7. **Performance Metrics**
8. **Challenges & Learnings:** Briefly discuss any technical challenges you faced and what your team learned during the hackathon.

UTD Career Guidance AI - Product Evaluation Rubric

1. Core Functionality (40 points)

Agent Performance & Output Quality

- **Excellent (32-40 pts):** Agents autonomously gather accurate data and provide highly relevant, personalized career guidance. Recommendations are specific, actionable, and clearly connected to real job market data.
- **Good (24-31 pts):** Agents function autonomously with mostly accurate data. Recommendations are relevant but may lack some specificity or depth.
- **Fair (16-23 pts):** Agents work but require manual intervention or produce generic recommendations with weak connections to job market reality.
- **Poor (0-15 pts):** Agents fail to work independently or provide irrelevant/inaccurate career guidance.

2. Agentic Architecture (15 points)

Agent Specialization & Coordination

- **Excellent (12-15 pts):** Clear separation of agent responsibilities using AWS Bedrock AgentCore. Agents coordinate seamlessly, demonstrating true autonomous collaboration. Each agent adds distinct value.
- **Good (9-11 pts):** Agents have defined roles and coordinate effectively, though some overlap or inefficiencies exist.
- **Fair (6-8 pts):** Basic agent structure exists but coordination is manual or limited. Unclear why multiple agents are needed.
- **Poor (0-5 pts):** Single-agent system disguised as multi-agent, or agents don't meaningfully interact.

3. Product Value & Usability (20 points)

Student-Centric Design

- **Excellent (16-20 pts):** Solves a real student pain point in an intuitive way. Interface is conversational and easy to use. Students would genuinely want to use this product.
- **Good (12-15 pts):** Addresses student needs with a functional interface. Some friction points but overall valuable.
- **Fair (8-11 pts):** Technically functional but doesn't feel like a product students would choose to use. Poor UX or unclear value proposition.
- **Poor (0-7 pts):** Confusing interface or fails to address stated problem. Feels like a demo rather than a product.

4. Technical Implementation & Deployment (15 points)

AWS Services Integration & Hosting

- **Excellent (12-15 pts):** Sophisticated use of AWS Bedrock AgentCore with proper hosting. Live demo is stable and accessible. Demonstrates understanding beyond basic tutorials.
- **Good (9-11 pts):** Proper use of Bedrock services with functional deployment. Minor hosting/stability issues.
- **Fair (6-8 pts):** Basic AWS integration that works but doesn't leverage agentic features meaningfully. Deployment issues present.
- **Poor (0-5 pts):** Minimal or improper use of required AWS services. No working live demo.

5. Strategic Thinking & Documentation (10 points)

Problem Solving Approach & Communication

- **Excellent (8-10 pts):** Clear rationale for design decisions in report and video. Team articulates trade-offs and explains why their approach best serves UTD students. Documentation is thorough and professional.
- **Good (6-7 pts):** Reasonable explanations for key decisions with good documentation. Some product thinking evident.
- **Fair (4-5 pts):** Basic justification and documentation provided but lacks depth or product perspective.
- **Poor (0-3 pts):** Unable to explain why decisions were made or focused only on technical features. Poor documentation.

Starts: Monday, October 13 | 1:00 PM

Ends (Submissions Due): Thursday, October 23 | 12:00 AM (Midnight)

Details: Following the workshop, your team will have 10 days to collaborate, innovate, and bring your autonomous AI agent to life using AWS Bedrock. This is your chance to apply your skills to a real-world challenge.

Deliverables

- **Project report (PDF/docx)**
- **Public GitHub repository with code**
- **Live link (hosted on AWS)**
- **3-5 minute video demo**
- **README with setup instructions**

Teams missing any required deliverable may receive reduced scores in relevant categories.