

Machine Intellect Society - Team Recruitment Tasks

Overview

As part of the recruitment process for the Machine Intellect Society - The AI Student Club of KLS GIT, candidates will complete tasks in two phases:

1. **Task Phase**
2. **Personal Interview Phase**

The tasks provided in this document will evaluate candidates' skills in Web Development, Machine Learning, and AI integration. Each task has specific requirements and must be deployed and hosted for accessibility from the club's systems. There are no restrictions on the use of AI tools or any other necessary technologies.

Task Options

Task 1: Simplified Document Query System

Objective: Build a basic document query system where users can upload a document and get relevant answers based on their queries.

Requirements:

- Use pre-built libraries like **LangChain** or **Haystack** to implement the document query feature.
- Limit to a single-agent system (no multi-agent functionality required).
- Use a basic UI framework like **Streamlit** instead of designing a custom web UI.
- Focus on handling text-based documents (e.g., .txt, .pdf).

Bonus Points:

- Allow users to see query history.
 - Implement exportable results for answers.
-

Task 2: Simple ML Problem

Objective: Build a model to classify handwritten digits using the MNIST dataset.

Requirements:

- Use pre-processed data from **scikit-learn** or **Keras datasets** (no need for complex preprocessing).
- Train basic ML models (e.g., Logistic Regression, Decision Trees) before introducing more advanced options.
- Use a simple web app framework (e.g., Streamlit or Flask) for deploying the solution.
- Provide an interactive interface to input digit images and see predictions.

Bonus Points:

- Visualize the classification process with a heatmap or similar.
 - Add a feature to show model accuracy for each digit class.
-

Task 3: Basic Landing Page

Objective: Create a static landing page for the Machine Intellect Society.

Requirements:

- Design an attractive and responsive page that includes:
 - Club name and logo.
 - Mission and vision statements.
 - Contact information and social media links.
- Use static hosting platforms like **GitHub Pages** or **Netlify** for deployment.
- Ensure cross-browser compatibility and mobile responsiveness.

Bonus Points:

- Add animations or interactive elements (e.g., carousels, scroll effects).
 - Use dynamic data (e.g., fetch and display upcoming events from a JSON file).
-

Submission Guidelines

1. Complete and deploy the assigned task.
2. Share the hosted link and source code repository in the Google Form that will be released.

3. A video demonstrating your project, to be included in the same Google Form as above.
4. Include a short document (PDF) explaining:
 - How the task was implemented.
 - Challenges faced and how they were resolved.
 - Tools and technologies used.

Deadline

For 2nd and 6th Semester: 27th March 2025

For 4th Semester: 31st March 2025

Evaluation Criteria

- **Functionality:** How well does the solution meet the task requirements?
 - **Creativity:** Novelty in design and approach.
 - **Technical Proficiency:** Effective use of tools and technologies.
 - **Deployment:** Accessibility and seamless operation of the deployed solution.
 - **Documentation:** Clarity and completeness of the accompanying report.
-

Good luck! We look forward to seeing your innovative solutions.