Judgment Criteria for OlympAI Hackathon

1. Technical Difficulty and Complexity (30%)

- **Code Quality (15%)**:
  - How clean, efficient, and well-structured is your code?
  - Is the code modular, maintainable, and scalable?
  - Are there clear and concise comments explaining the logic, especially in complex sections of the code?
  - Is the use of data structures and algorithms appropriate for the problem?
- **Implementation Depth (15%)**:
  - How difficult is the problem you are solving?
  - Does the solution involve advanced concepts of AI, such as machine learning, neural networks, or deep learning models?
  - Have you incorporated AI techniques or algorithms effectively?
  - Does the code demonstrate in-depth technical knowledge and understanding?

2. Innovation (20%)

- **Novelty of the Idea (10%)**:
  - How unique and creative is the solution?
  - Does it offer a new approach to solving an existing problem?
  - Is there any original thought process or unconventional methods used in the solution?
- **Creative Use of AI (10%)**:
  - Are AI tools and techniques used in innovative ways?
  - Is there a fresh perspective or experimentation with new AI technologies or methodologies?

3. Impact and Sustainability (20%)

- **Real-world Relevance (10%)**:
  - How well does the project address a real-world problem or societal challenge?
  - Can the solution be scaled up for broader use, and does it have potential long-term benefits?
  - Is the problem tackled with a meaningful solution that can bring positive change?
- **Sustainability and Feasibility (10%)**:
  - Is the project sustainable over time in terms of cost, resource use, and practicality?
  - Can the solution continue to operate and improve beyond the initial implementation?
  - Does it have the potential for future development or integration into real-world systems?

4. User Experience (15%)

- **Ease of Use (10%)**:
  - Is the solution easy to understand and use for its target audience?
  - How intuitive and user-friendly is the interface (if applicable)?
  - Are there clear instructions for running the code or using the tool?
- **Accessibility and Inclusivity (5%)**:
  - Does the solution consider accessibility for diverse users, including those with different abilities or technological access?
  - Does it promote inclusivity and ensure ease of access for its intended audience?

5. Presentation and Communication (15%)

- **Video Demonstration (10%)**:
  - Does the YouTube video clearly and effectively demonstrate the functionality of the project?
  - Is the problem being solved clearly articulated, and is the approach well-explained?
  - Are the key features of the solution highlighted in an engaging and concise manner?
  - Is the overall quality of the video, including visuals and narration, professional and easy to follow?
- **Project Presentation (5%)**:
  - Does the PowerPoint presentation clearly explain the project's goals, methodologies, and results?
  - Is the presentation structured logically with relevant visuals, diagrams, or screenshots?
  - Does the student effectively communicate their thought process, problem-solving approach, and key takeaways?

Submission Guidelines

- **Code File**:
  - Submit a link to your code file via a GitHub repository or similar platform. Ensure that the code is well-commented for readability and that external dependencies or setup instructions are clearly documented.
- **YouTube Video**:
  - Upload a video demonstrating your project. It should clearly explain the problem you're solving, how the solution works, and how it addresses the challenge. Ensure that the video showcases the key aspects of your project, including demonstrations of the AI techniques used.
- **Project Presentation (PPT)**:
  - Submit a detailed PowerPoint presentation covering all aspects of your project: problem definition, solution overview, AI techniques used, impact, user experience, and sustainability. The presentation should be concise but detailed enough to communicate your approach effectively.