**Opening:** “Hello, I’m Josiah from Team Weka. This is the presentation of our project: *‘A Holistic Approach to Integrating AI in Government Document Workflows.’* This project addresses the challenge “Using AI to transform bureaucratic jargon into plain English.”

The goal of our project is to modernize and streamline government document workflows using AI. I’ll talk about our approach in six sections, each focusing on different aspects of AI integration. These techniques help ensure accuracy, security, and efficiency in creating and managing government content.”

**Body:**

**1. Platform Architecture and Technology Selection:** “Firstly: Platform architecture. We determined a mix of cloud-based and on-premise solutions would be best, depending on if the task has security requirements. For sensitive tasks that require enhanced security, we can deploy Meta’s Large Language Model AI, or **LLaMA,** on local servers using **NVIDIA GPUs**. This setup ensures that the AI operates within an isolated environment, with **load balancing** to manage high workloads. However, for general content tasks, we can use cloud-based technology. **Azure OpenAI Service** provides powerful **Natural Language Processing** capabilities and integrates well with other tools. “

**2. Data Collection and Annotation Process:** “Our second concern is collecting and annotating data. **Scrapy** and **BeautifulSoup** can be used to scrape text from government websites and policy databases. This ensures we capture up-to-date, relevant information. After collecting the data, we clean it using **Natural Language Processing tools like NLTK and SpaCy** to remove irrelevant or redundant information. Finally, we use manual annotation platforms such as **Amazon Mechanical Turk** to label key attributes like grammar, tone, and clarity, ensuring the dataset is aligned with the Australian Government Style Manual.”

**3. Model Fine-Tuning and Optimization:** “Our third section is about fine-tuning the AI models. Here, we use **multi-task learning architectures** such as **T5** and **BART**, which allow the AI to handle multiple tasks like spell-checking, sentence simplification, and tone adjustment simultaneously. To optimize these models, we apply **Transformer optimization techniques** like **sparse attention**, which reduce the computational burden. In cases where devices have limited resources, we use **knowledge distillation** to compress the model’s size, ensuring it can still operate effectively on lightweight devices.”

**4. Real-Time Monitoring and Continuous Improvement:** “Our fourth section is about monitoring and continuous improvement. Real-time monitoring systems using **Prometheus** and **Grafana** are able to track key performance metrics such as latency and response time. As the system is maintained, **Continuous Integration & Deployment pipelines** allow testing of new model versions and automatic deployment of updates. Rollback mechanisms are also vital. When degradation occurs, it can revert to a stable version, ensuring that the system remains reliable.”

**5. Ensuring Compliance with Australian English:** “Our fifth section addresses the AI’s ability to adhere to Australian English conventions. We expand the AI’s language dataset by collecting content from government archives and Australian media. To ensure the AI uses correct spelling and grammar, we implement **dynamic spell-checking algorithms** like **Hunspell**, which adjust spelling based on the context. The AI will consistently follow Australian English rules, thanks to the use of these tools in combination with the Australian Government Style Manual.

**6. Retrieval-Augmented Generation (RAG) for Accuracy:** “Finally, we can ensure that the AI generates accurate content using **Retrieval-Augmented Generation or RAG** models. With RAG, background information is retrieved from government databases before generating content. **BERT embeddings** are used to retrieve relevant data, which is then integrated into the generation process. We also advise implementing a **generate-retrieve-verify loop** where AI-generated content is compared to retrieved data. If any discrepancies arise, the content is regenerated to ensure it matches the source.”

**Closing:** “In conclusion, in our approach we’ve considered everything from platform selection to ensuring accuracy and compliance with Australian standards. By using a combination of cloud-based and on-premise AI solutions, advanced data collection, and model optimization techniques, we are able to modernize government document workflows. This ensures not only high-quality content but also efficiency and accessibility for all users. This concludes the presentation, thank you for listening.”

### **Summary of Sections and Techniques:**

1. **Platform Architecture**:
   * **Techniques**: Azure OpenAI Service, LLaMA, NVIDIA A100 GPUs, Load Balancing.
2. **Data Collection & Annotation**:
   * **Techniques**: Scrapy, BeautifulSoup, NLTK, SpaCy, Amazon Mechanical Turk.
3. **Model Fine-Tuning & Optimization**:
   * **Techniques**: T5, BART, Sparse Attention, Knowledge Distillation.
4. **Real-Time Monitoring & Continuous Improvement**:
   * **Techniques**: Prometheus, Grafana, CI/CD Pipelines.
5. **Australian English Compliance**:
   * **Techniques**: Dynamic Spell-Checking (Hunspell), Localized Datasets.
6. **Retrieval-Augmented Generation (RAG)**:
   * **Techniques**: BERT Embeddings, Generate-Retrieve-Verify Loop.

Condensed Version to fit into 3 minutes

**Opening:** “Hello, I’m Josiah from Team Weka. Our project addresses the GovHack challenge “Using AI to transform bureaucratic jargon into plain English.”

I’ll talk about our approach in six sections, each focusing on different aspects of AI integration.”

**Body:**

**1. Platform Architecture and Technology Selection:** “Firstly: Platform architecture. We decided on a mix of cloud-based and on-premise solutions. For sensitive tasks, we can deploy Meta’s Large Language Model AI on local servers.

However, for general tasks, we can use cloud-based technology. **Azure OpenAI Service** provides powerful **Natural Language Processing** capabilities and integrates well with other tools. “

**2. Data Collection and Annotation Process:** “Secondly: Collecting and annotating data. **Scrapy** and **BeautifulSoup** can capture up-to-date, relevant information from government databases. Then, **Natural Language Processing tools like NLTK** can remove irrelevant or redundant information. Finally, **Amazon’s Mechanical Turk** annotation platform key attributes to ensure the dataset is aligned with the Government Style Manual.”

**3. Model Fine-Tuning and Optimization:** “Thirdly: fine-tuning the AI models requires **multi-task learning architectures** such as **T5** and **BART**. To optimize these models, we apply **Transformer optimization techniques** like **sparse attention**. **Knowledge distillation** is also used to compress the model’s size, ensuring it can operate on lightweight devices.”

**4. Real-Time Monitoring and Continuous Improvement:** “Our fourth section is about monitoring and continuous improvement. Systems using **Prometheus** are able to track key performance metrics such as latency in real-time. As the system is maintained, **Continuous Integration & Deployment pipelines** allow testing of new versions and automatic updates. As a backup, Rollback Mechanisms will ensure the system is always accessible.

**5. Ensuring Compliance with Australian English:** “Our fifth section addresses the AI’s adherence to Australian English. By collecting content from media and government, we expand the language dataset. And by implementing **dynamic spell-checking algorithms** like **Hunspell**, we ensure the AI will follow correct spelling and grammar regardless of context.

**6. Retrieval-Augmented Generation (RAG) for Accuracy:** “Finally, we can ensure that the AI generates accurate content using **Retrieval-Augmented Generation or RAG** models. With RAG, background information is retrieved before generating content. This is then integrated into the generation process. We also advise implementing a **generate-retrieve-verify loop** where generated content is compared to retrieved data. That way, the content is regenerated until it matches the source.

Thank you for listening.”