**Title –** Exploring OpenAI Playground

**Concept** - AI-powered tools to generate text, stories, and code using prompts**.**

**Theory applied to experiment-** Students will learn how prompts in OpenAI Playground can be used to generate code, creative stories, and more. They will explore how AI can assist with coding tasks without needing prior coding experience.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Materials required** | **Hardware Requirements:**   |  |  | | --- | --- | | **Components** | **Quantity** | | Computer or laptop with internet access | 1 |   **Software Requirements:**   |  |  | | --- | --- | | **Components** | **Quantity** | | **Links:** [OpenAI Playground](https://platform.openai.com/playground) | 1 | | Web browser (Google Chrome recommended) | 1 | |
| **Procedure** | **Refer this link**  **YouTube link-** [**https://youtu.be/iwYtzPJELkk?si=wisUvGZQ12qSkL7R**](https://youtu.be/iwYtzPJELkk?si=wisUvGZQ12qSkL7R) |
| **Procedure to be performed by student** | **Procedure to be performed by student:** **OpenAI Components**  1. **Temperature**    * **What it does**: Controls creativity.    * **Simple**: Higher = more creative, lower = more predictable. 2. **Max Tokens**    * **What it does**: Limits response length.    * **Simple**: Higher = longer answers, lower = shorter answers. 3. **Top P**    * **What it does**: Chooses safe or creative responses.    * **Simple**: Lower = safe answers, higher = more creative answers. 4. **Frequency Penalty**    * **What it does**: Stops repetition.    * **Simple**: Higher = less repeating of words. 5. **Presence Penalty**    * **What it does**: Adds new ideas.    * **Simple**: Higher = more new ideas, lower = stays on the same topic.   **Procedure to Perform**:   1. **Access OpenAI Playground**:    * Open the link to the [OpenAI Playground](https://platform.openai.com/playground) in your browser.    * Sign up or log in to your account if required. 2. **Start a New Session**:    * Click on "Create new playground" to start your project. 3. **Enter Prompt**:    * For each use case, type a specific prompt (e.g., "Generate Python code to create a basic calculator").    * Press "Submit" to generate the result. 4. **Customize**:    * Review the output generated by the AI in response to your prompt.    * Modify or copy the generated code if needed, and test it in another environment (e.g., Jupyter Notebook). 5. **Save or Try New Prompts**:    * Copy the generated code, text, or story to a separate document or coding environment.    * Try other prompts to see different results.   **Use Case 1: Simple Calculator Code**  **Prompt**: "Write Python code to create a simple calculator that can add, subtract, multiply, and divide."  **Steps**:  **1. Open OpenAI Playground:**   * Access **OpenAI Playground** by visiting <https://platform.openai.com/playground>. * Sign in using your credentials or create an account if needed.   **2. Input the Prompt:**   * In the text box of OpenAI Playground, type this prompt:   “Write Python code to create a simple calculator that can add, subtract, multiply, and divide.”  **3. Generate the Code:**   * Click the **Submit** or **Generate** button to get the Python code for the calculator. * Once the code is generated, **copy** the entire code.   **4. Open Jupyter Notebook:**   * On your computer, open **Jupyter Notebook**:   **5. Create a New Notebook:**   * In the Jupyter dashboard, click the **New** button on the top right and select **Python 3**. * A new notebook will open with an empty code cell.   **6. Paste the Calculator Code:**   * In the first code cell of the notebook, **paste** the Python code that was generated in OpenAI Playground.   **7. Run the Code:**   * To execute the code, click the **Run** button (or press **Shift + Enter**). * The code will run, and you’ll be prompted to enter numbers and choose an operation like addition or subtraction.   **8. Interact with the Calculator:**   * After running the code, you will be prompted to:   + **Enter the operation** (e.g., "1" for addition, "2" for subtraction).   + **Enter two numbers** for the calculation. * The result of your chosen operation (addition, subtraction, multiplication, or division) will be displayed right after.   **9. Save the Notebook:**   * Once you’ve successfully run and tested the calculator, click **File** → **Save As** to save your notebook.   **Use Case 2: Generate a Fun Poem**  **Prompt**: "Write a fun poem about a cat who loves pizza."  **Steps**:   1. Start a new session in OpenAI Playground. 2. Input the poem prompt in the text box. 3. Review the generated poem and make any edits if you'd like to personalize it. 4. Share the poem with your classmates, or illustrate the poem by drawing pictures to go along with it.   **Use Case 3: Generate Code for a Simple Rock-Paper-Scissors Game**  **Prompt**: "Write Python code for a simple rock-paper-scissors game that a player can play against the computer."  **Steps**:   1. Start a new session in OpenAI Playground. 2. Input the rock-paper-scissors game prompt in the text box. 3. Review the generated Python code, and copy it into your IDE or Notebook. 4. Run the code, play the game, and try modifying the code (e.g., add a score counter). 5. Share your modified version of the game with the class |
| **Observation /analysis** |  Students will observe how the AI tool generates outputs based on different prompts.   They will analyze how AI can write code, create fun content like poems, and generate interactive games. |
| **Conclusion** | OpenAI Playground is a powerful tool for more than just text—it can help generate creative writing, code, and even interactive games. This experiment shows students how AI simplifies complex tasks, making coding and writing both fun and accessible. |
| **Experiment takeaways** | * AI can generate code, stories, and games from simple prompts. * Students can use OpenAI Playground to experiment with different creative and coding tasks. * AI tools like OpenAI encourage creativity and learning by reducing the complexity of technical tasks. |
|  |  |