LABORATORIUM

Optimalkan kode yang dihasilkan AI menggunakan model IBM Granite

IBM **SkillsBuild**

Daftar Isi

Pendahuluan

Lab ini berfokus pada pengoptimalan kode yang dihasilkan AI menggunakan teknik prompt few-shot dengan model IBM Granite di Google Colab untuk mengoptimalkan kualitas pembuatan kode. Prompt few-shot memungkinkan model menggambar dari beberapa contoh, sehingga memungkinkan hasil yang lebih akurat dan bernuansa dibandingkan dengan pendekatan prompt zero-shot awal.

Lab ini dibangun berdasarkan lab 1, yang berfokus pada pembuatan kode menggunakan model Granite, dengan mengalihkan fokus untuk menyempurnakan keluaran model melalui teknik prompt few-shot. Lab ini berfokus pada pengoptimalan komponen UI untuk membuat antarmuka yang lebih intuitif untuk toko buku online, mendemonstrasikan bagaimana menggabungkan contoh dapat meningkatkan kualitas dan ketepatan kode yang dihasilkan.

Di lab ini, kamu akan menggunakan rangkaian model IBM Granite yang dihosting di Replicate untuk mengoptimalkan kode Python yang menyempurnakan desain halaman arahan untuk toko buku online.

Persyaratan perangkat lunak

Untuk menyelesaikan lab ini, kamu harus memiliki akses ke akun Replicate untuk melakukan panggilan inferensi model. Kamu juga harus mendapatkan token API dari akun Replicate, yang akan ditambahkan dengan aman sebagai rahasia Google Colab.

Pemahaman tentang Python tidak diperlukan, tetapi dapat membantu untuk memahami kode yang dihasilkan.

Tujuan

Setelah menyelesaikan lab ini, kamu seharusnya dapat:

- Mengoptimalkan kode yang dihasilkan AI menggunakan model IBM Granite Code

Langkah-langkah lab

Lab ini mengharuskan kamu untuk menyelesaikan langkah-langkah berikut:

- Langkah 1: Mengonfirmasi pengaturan perangkat lunak
- Langkah 2: Memuat Jupyter notebook dan menginisialisasi model
- Langkah 3: Menghasilkan kode dengan menggunakan prompt few-shot

- Langkah 4: Mengoptimalkan kode yang dihasilkan AI

Perkiraan durasi untuk menyelesaikan
30 menit

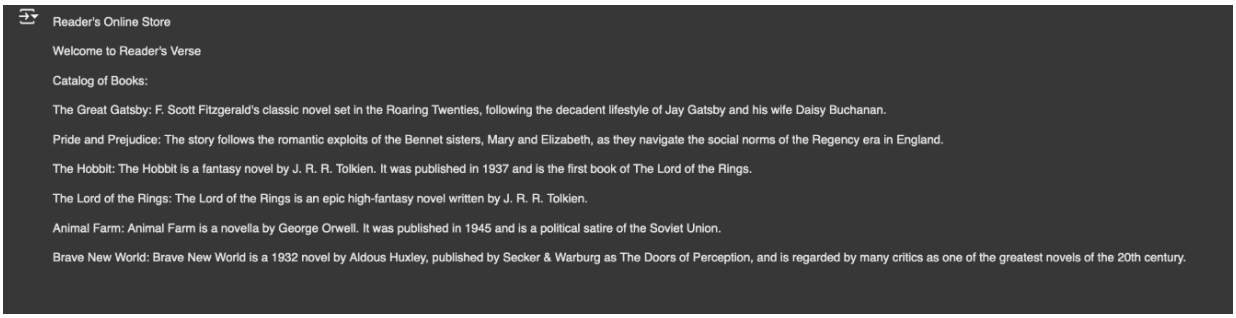
Skenario

Latar belakang

Reader's Verse adalah toko buku lokal yang berencana membangun kehadiran online dengan membuat situs web yang memungkinkan para pembacanya untuk mencari katalog dan memeriksa ketersediaan sebelum mengunjungi toko fisik. Kamu adalah perancang web yang ditugaskan untuk proyek ini dan telah membuat draf awal situs web dengan menggunakan teknik prompting zero-shot.

Tantangan

Klien menghargai hasil awal dan meminta penyempurnaan untuk meningkatkan desain visual dan fungsionalitas situs web secara keseluruhan. Kamu akan menggunakan model IBM Granite untuk mengoptimalkan kode yang dihasilkan AI dari Lab 1 untuk menyempurnakan desain awal situs web.



Solusi

Kamu akan menggunakan teknik prompt few-shot untuk mengoptimalkan kode yang dihasilkan AI menggunakan IBM Granite dan meningkatkan desain visual serta fungsionalitas situs web.

Langkah 1: Mengonfirmasi pengaturan perangkat lunak

Ikhtisar

Pada langkah ini, pastikan kamu telah menyelesaikan langkah-langkah berikut dari lab 1:

- Langkah 1: Membuat akun GitHub
- Langkah 2: Membuat akun Replicate
- Langkah 3: Mendaftar ke Google Colab

Jika kamu belum menyelesaikan langkah-langkah ini, lihat dokumen “Lab 1: Menggunakan model IBM Granite untuk pembuatan kode dan tugas pemrograman” untuk detailnya dan selesaikan langkah-langkahnya sekarang.

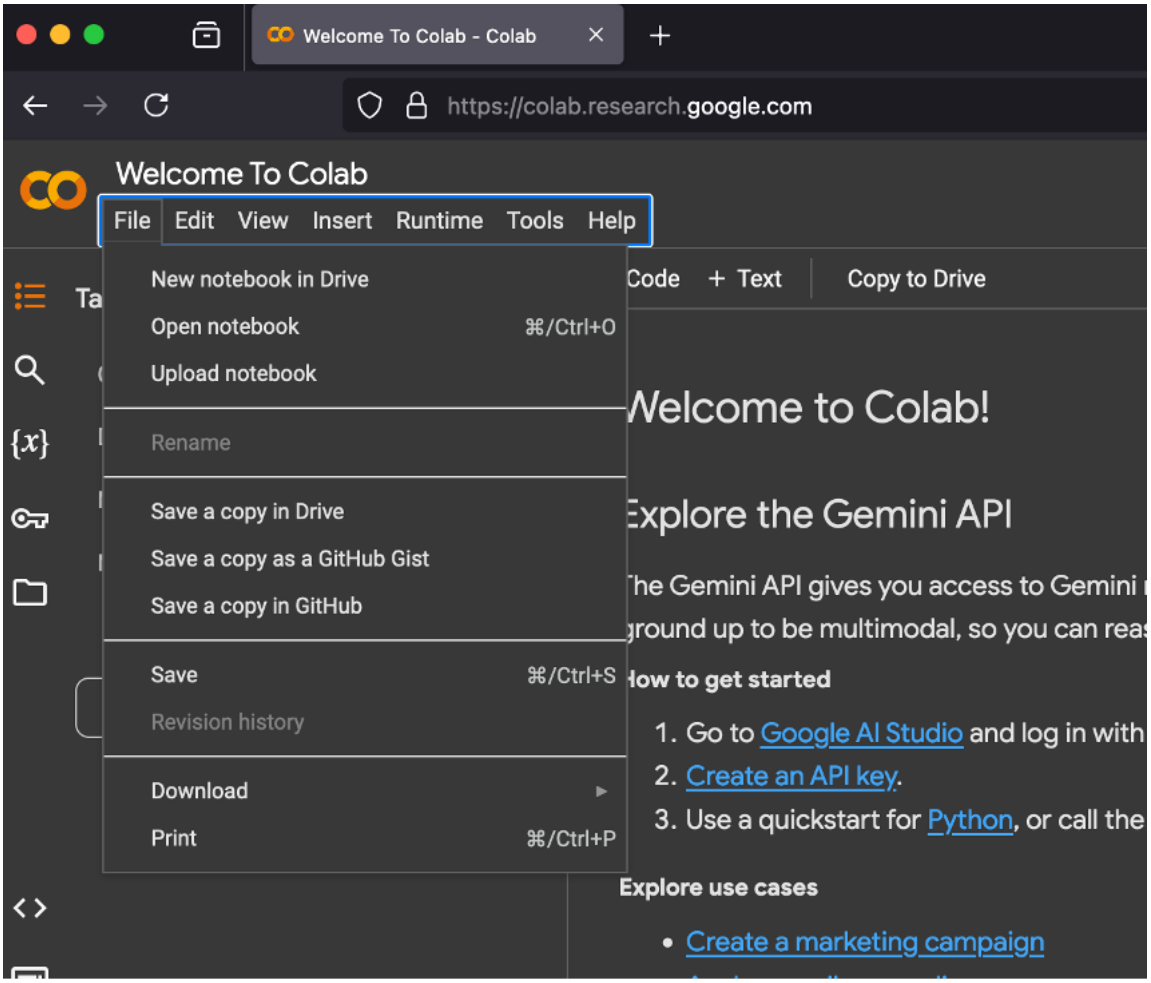
Langkah 2: Memuat Jupyter notebook dan menginisialisasi model

Ikhtisar

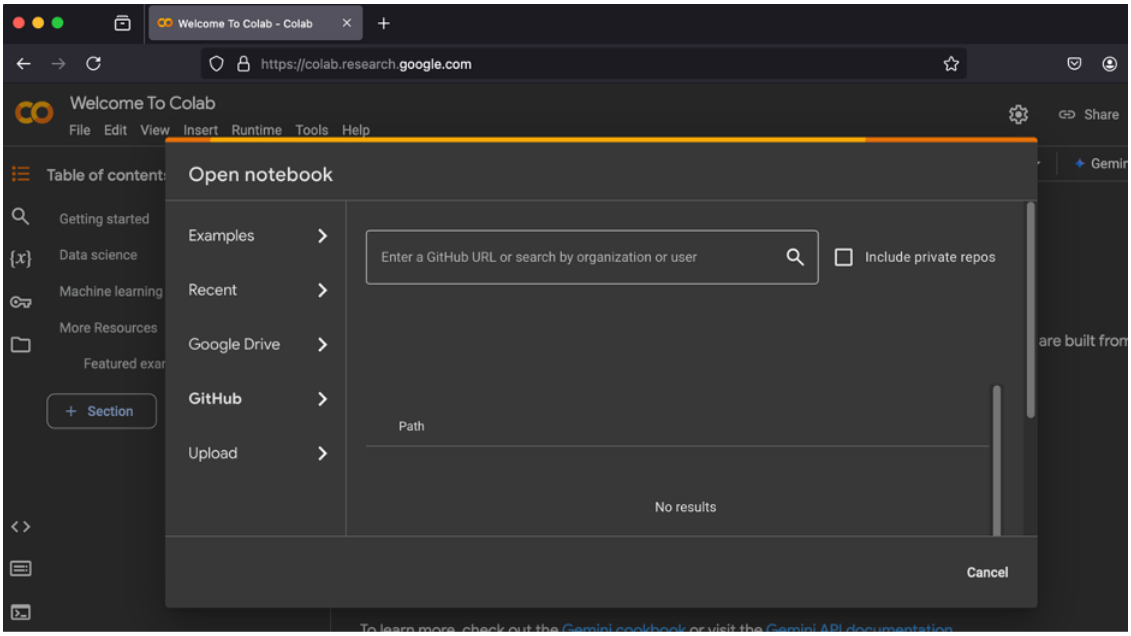
Pada langkah ini, kamu akan memuat Jupyter notebook yang berisi kode yang diperlukan untuk melakukan lab ini. Jupyter notebook adalah dokumen yang dapat dibagikan yang menggabungkan kode komputer, deskripsi bahasa sederhana, data, dan visualisasi. Kamu akan memuat file Jupyter notebook dari GitHub ke Google Colab dan menginisialisasi model IBM Granite untuk menghasilkan kode untuk skenario yang diberikan. Notebook berisi rincian prompt few-shot untuk mengoptimalkan output awal dari lab 1.

Instruksi

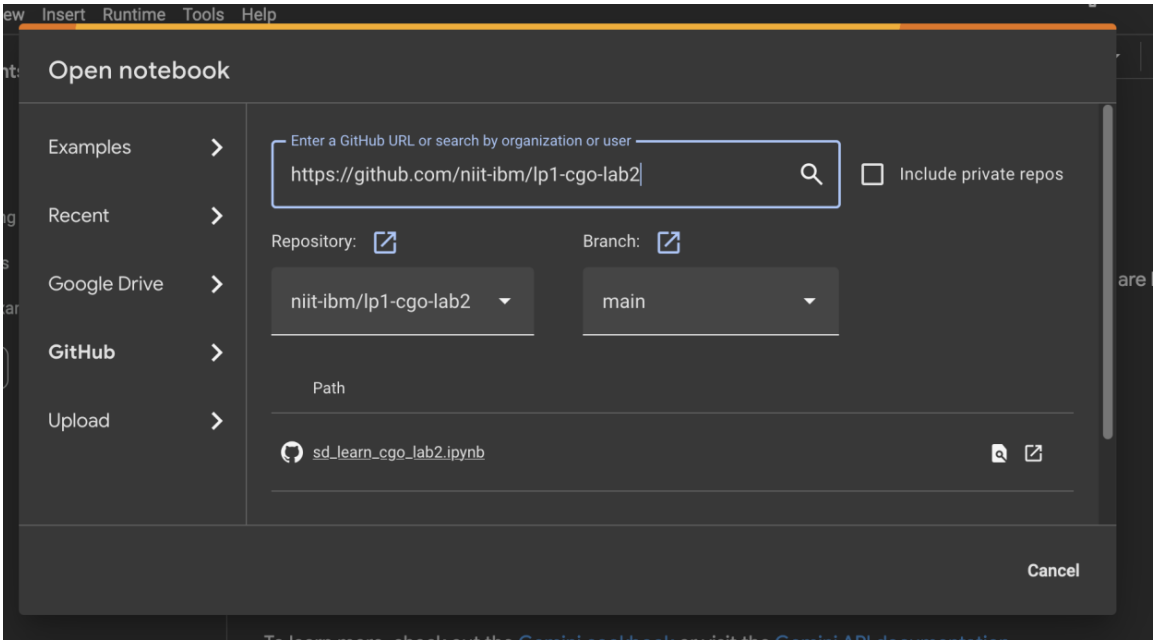
1. Di ruang kerja Google Colab, pada menu **File**, pilih **Open notebook (Buka notebook).**



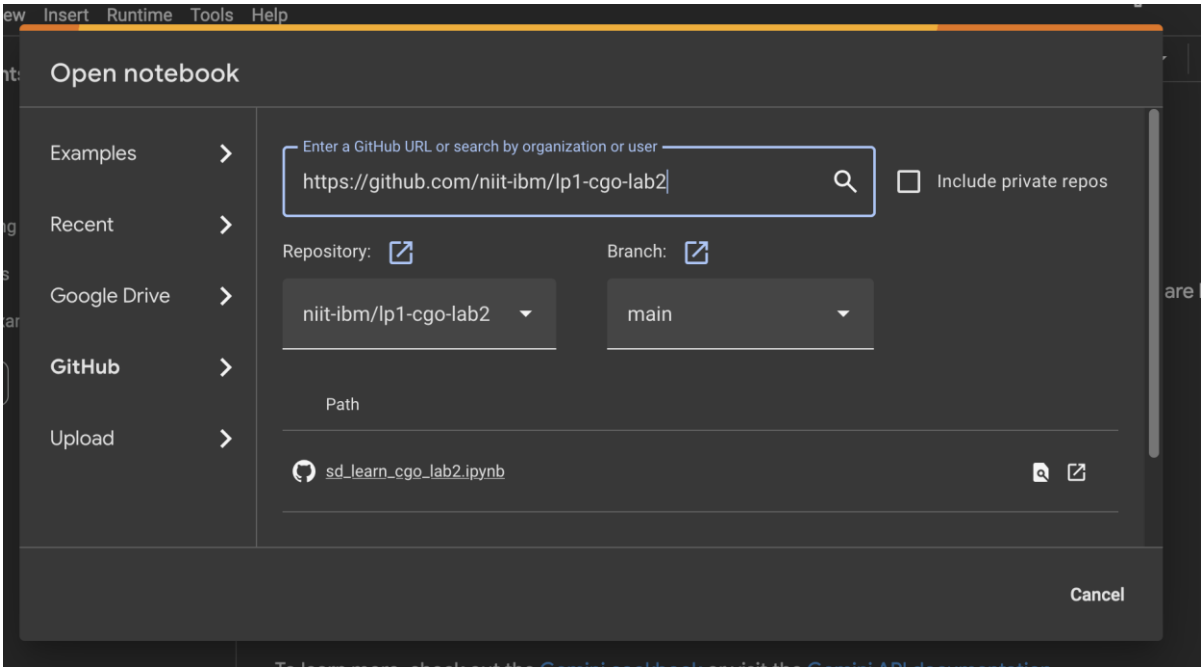
2. Pilih tab **GitHub**.



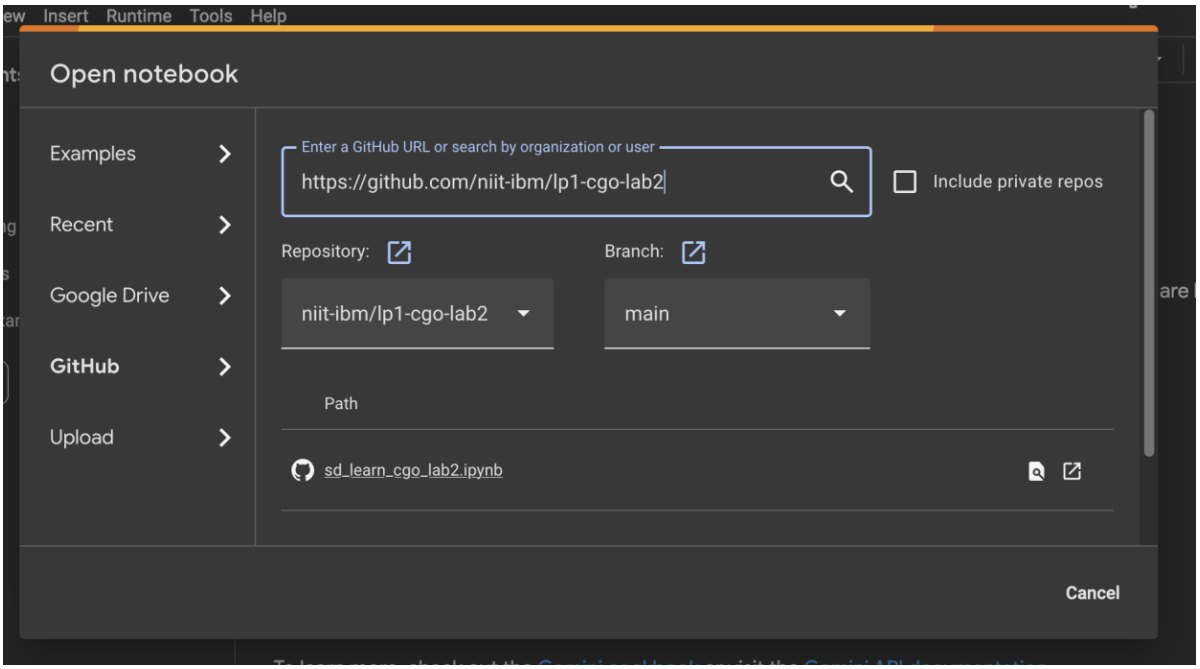
3. Pada bidang **Enter a GitHub URL or search by organization or user (Masukkan URL GitHub atau cari berdasarkan organisasi atau pengguna)**, salin URL berikut ini: https://github.com/niit-ibm/lp1-cgo-lab2 dan pilih ikon **kaca pembesar**.



4. Pilih **main (utama)** di bagian **Branch (Cabang)**.

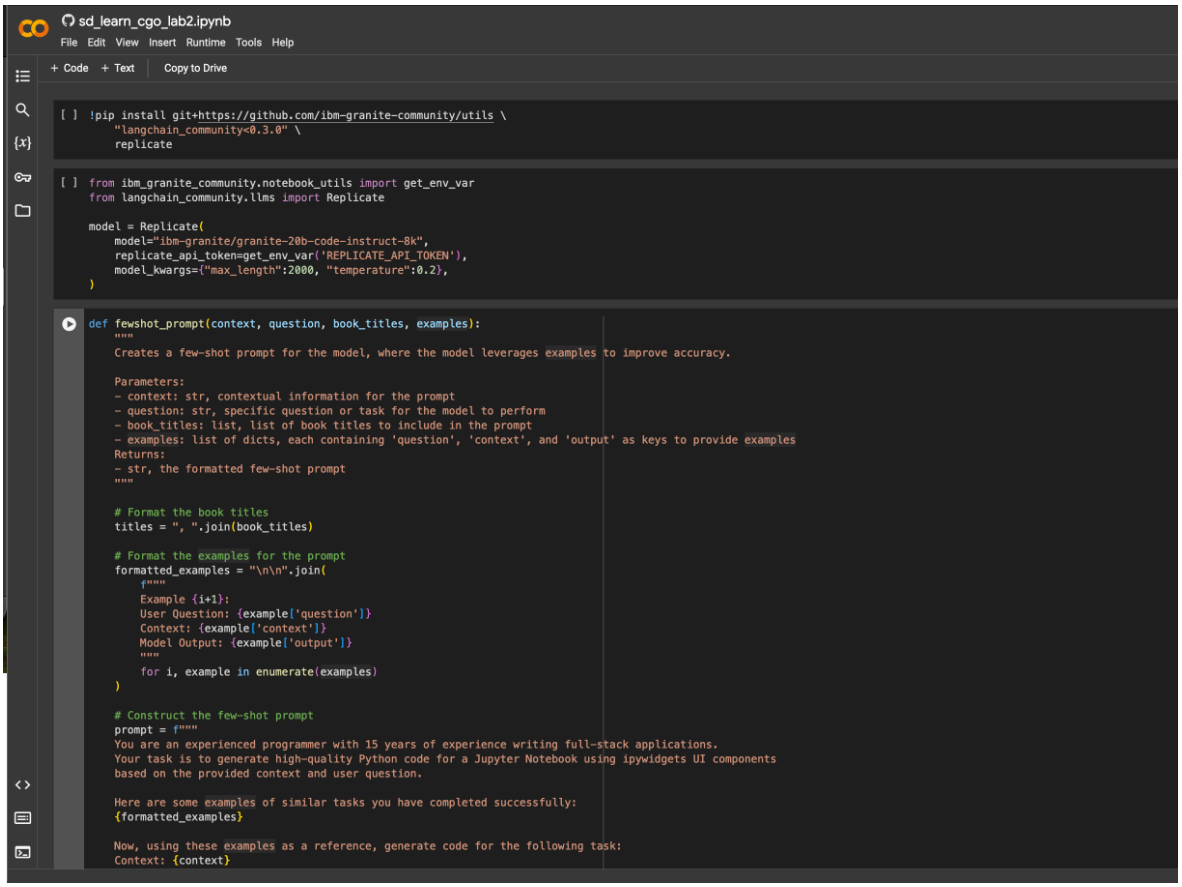


5. Pilih **sd_learn_cgo_lab2.ipynb** notebook di bagian **Path (Jalur)** untuk membuka notebook di Colab.



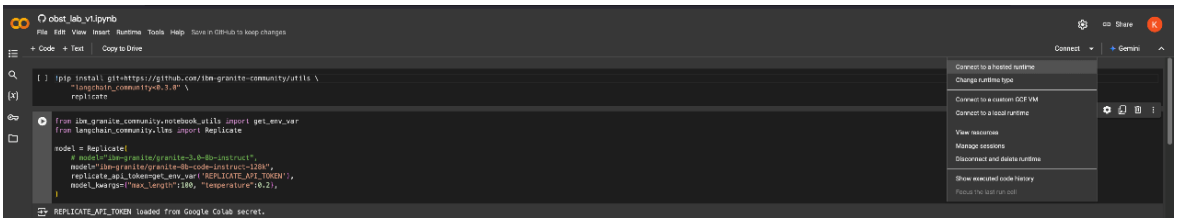
6. Kamu sekarang akan membuka notebook sd_learn_cgo_lab2 di ruang kerja Colab.

   Perhatikan bahwa setiap baris di notebook disebut sebagai **sel**. Sel-sel dalam notebook tidak diberi nomor. Notebook diatur sedemikian rupa sehingga kamu harus menjalankan kode di setiap sel secara berurutan, dimulai dari sel pertama, untuk menyelesaikan lab ini.

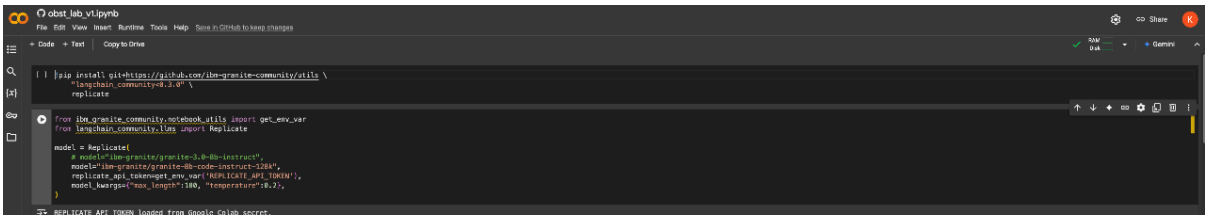


7. Kamu harus memulai contoh baru di runtime Google Colab untuk menjalankan kode di Jupyter notebook. Pilih **Connect (Hubungkan)** di bilah navigasi Google Colab.

8. Pilih **Connect to a hosted runtime (Hubungkan ke runtime yang dihosting)** pada menu **Connect (Hubungkan)**.



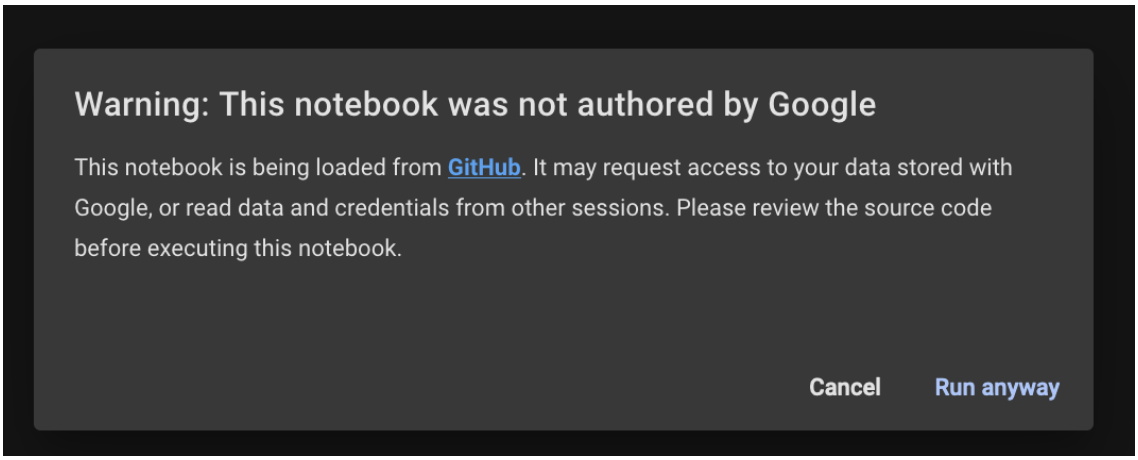
9. Tanda centang hijau menunjukkan kamu telah berhasil terhubung ke runtime yang dihosting.



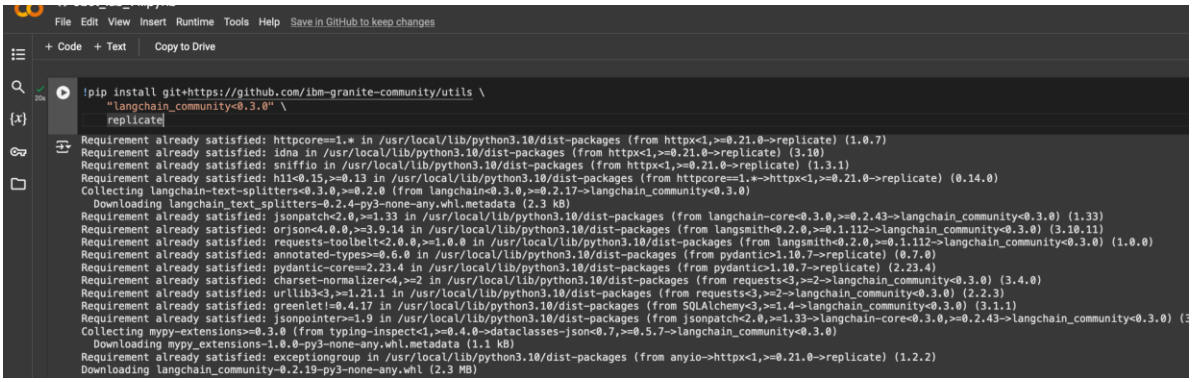
10. Di sel pertama notebook, pilih tombol **Putar** untuk menginstal pustaka yang diperlukan dari komunitas Granite.

Catatan: Kamu harus menjalankan sel secara berurutan, mulai dari sel pertama dalam notebook.

11. Pilih **Run anyway (Jalankan)** untuk melanjutkan pemuatan pustaka yang diperlukan.

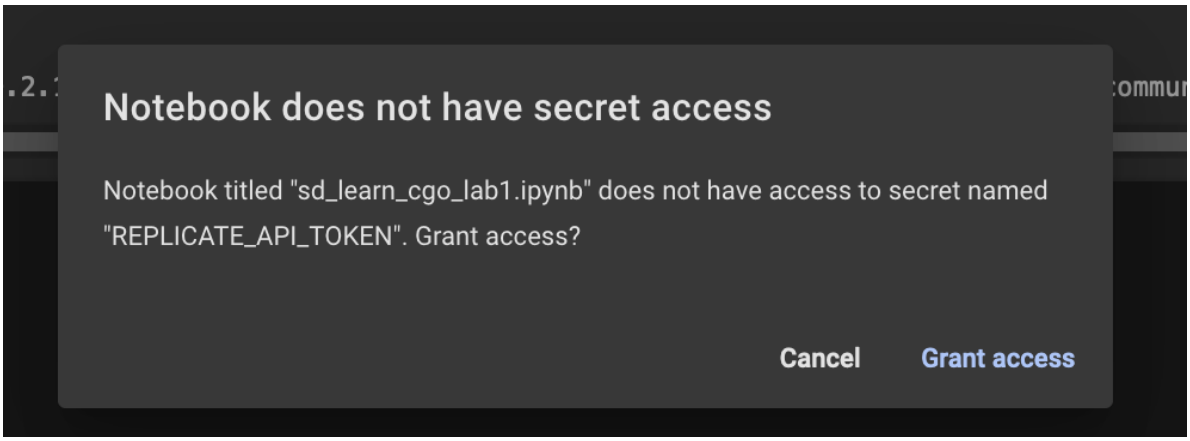


12. Tanda centang hijau muncul di samping tombol **Putar** untuk menunjukkan bahwa pustaka yang diperlukan berhasil diinstal.

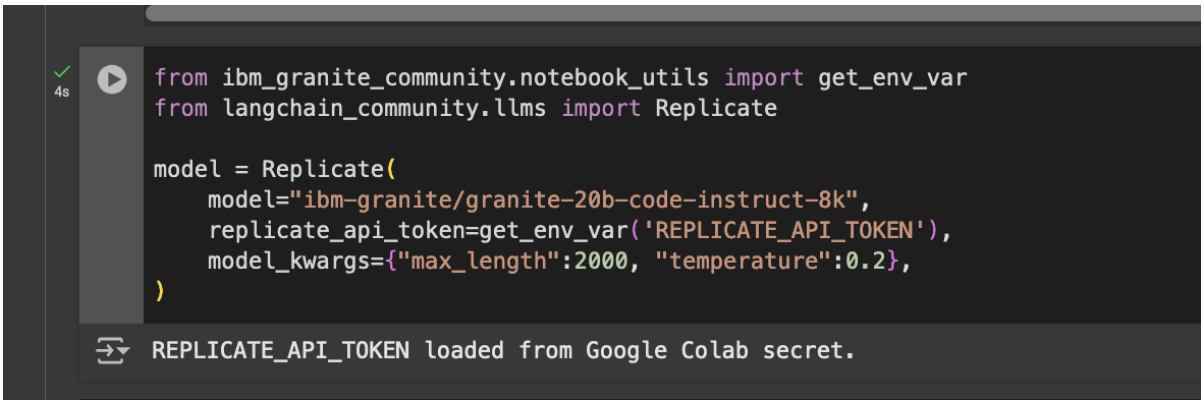


13. Pada sel kedua notebook, pilih tombol **Putar** untuk menginisialisasi model IBM Granite menggunakan Replicate untuk pembuatan kode.

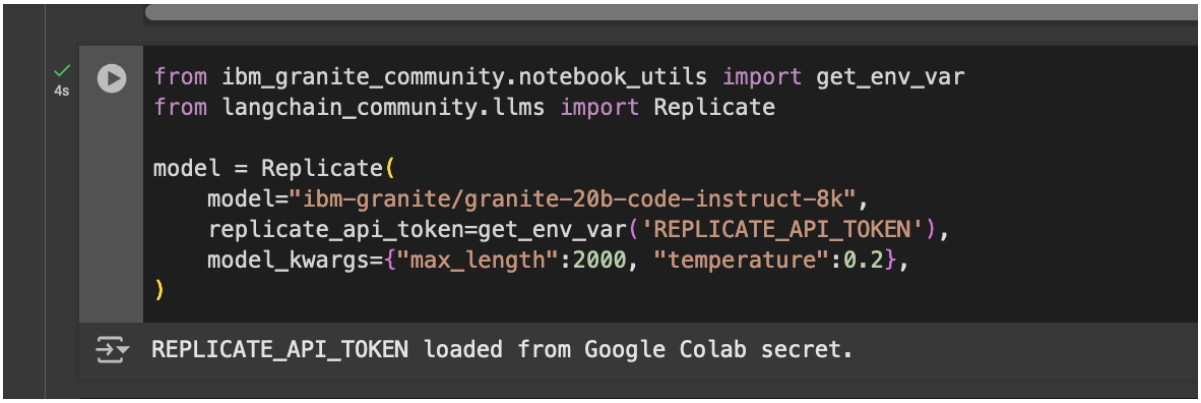
14. Pilih **Grant access (Berikan akses)** untuk melanjutkan pemuatan model dari Replicate.



15. Pesan berikut ditampilkan di bagian bawah sel: "REPLICATE_API_TOKEN dimuat dari rahasia Google Colab".



16. Tanda centang hijau di samping tombol **Putar** menunjukkan bahwa model IBM Granite diinisialisasi menggunakan Replicate. Kamu sekarang siap untuk meminta model menghasilkan kode untuk skenario yang diberikan.



Langkah 3: Menghasilkan kode dengan menggunakan prompt few-shot

Ikhtisar

Pada langkah ini, Kamu akan meminta model IBM Granite untuk menghasilkan kode untuk skenario yang diberikan dengan menggunakan petunjuk beberapa bidikan. Kamu akan menentukan prompt few-shot untuk model dan menjalankan prompt untuk menyempurnakan halaman arahan situs web Reader's Verse dengan menyertakan elemen interaktif.

Instruksi

1. Tentukan fungsi yang membuat beberapa instruksi untuk model AI menggunakan contoh untuk meningkatkan kualitas output. Fungsi ini akan membantu mengoptimalkan kode untuk membuat komponen UI interaktif untuk halaman arahan situs web Reader's Verse.

   Pada sel ketiga notebook, pilih tombol **Putar** untuk menentukan fungsi **fewshot_prompt**, yang berisi serangkaian instruksi yang memandu model pada output yang akan dihasilkan. Dalam skenario ini, instruksi menginformasikan model untuk mengoptimalkan UI untuk toko buku online berdasarkan parameter yang diberikan: konteks, pertanyaan, judul buku, dan contoh.

```python
def fewshot_prompt(context, question, book_titles, examples):
    """
    Creates a few-shot prompt for the model, where the model leverages examples to improve accuracy.

    Parameters:
    - context: str, contextual information for the prompt
    - question: str, specific question or task for the model to perform
    - book_titles: list, list of book titles to include in the prompt
    - examples: list of dicts, each containing 'question', 'context', and 'output' as keys to provide examples
    Returns:
    - str, the formatted few-shot prompt
    """

    # Format the book titles
    titles = ", ".join(book_titles)

    # Format the examples for the prompt
    formatted_examples = "\n\n".join(
        f"""
        Example {i+1}:
        User Question: {example['question']}
        Context: {example['context']}
        Model Output: {example['output']}
        """
        for i, example in enumerate(examples)
    )

    # Construct the few-shot prompt
    prompt = f"""
    You are an experienced programmer with 15 years of experience writing full-stack applications.
    Your task is to generate high-quality Python code for a Jupyter Notebook using ipywidgets UI components
    based on the provided context and user question.

    Here are some examples of similar tasks you have completed successfully:
    {formatted_examples}

    Now, using these examples as a reference, generate code for the following task:
    Context: {context}
    User Question: {question}
    Include descriptions of elements from the book titles: {titles}
    Ensure that:
    - The code is well-structured and uses appropriate styling, coloring, and formatting for UI elements.
    - Ensure the `value` strings in all `widgets.HTML` components use triple double quotes for multi-line HTML.
    - The output code is optimized and does not exceed 300 tokens.
    - Output only the Python code.
    """
    return prompt
```

Catatan: Kamu tidak akan melihat output apa pun apabila kamu menjalankan sel ini di notebook.

2. Berikutnya, tentukan fungsi untuk menghasilkan respons dari model dengan menggunakan prompt few-shot. Fungsi ini mengirimkan instruksi yang ditentukan dalam prompt few-shot ke model dan mendapatkan respons yang diinginkan dari model.

   Di sel keempat notebook, pilih tombol **Putar** untuk menentukan fungsi **get_answer_using_fewshot.**

```python
# Function to get answer using few-shot prompting
def get_answer_using_fewshot(context, question, book_titles, examples):
    """
    Generates the response from the model based on a few-shot prompt.

    Parameters:
    - context: str, contextual information for the prompt
    - question: str, specific question for the model to answer
    - book_titles: list, list of book titles to include in the prompt

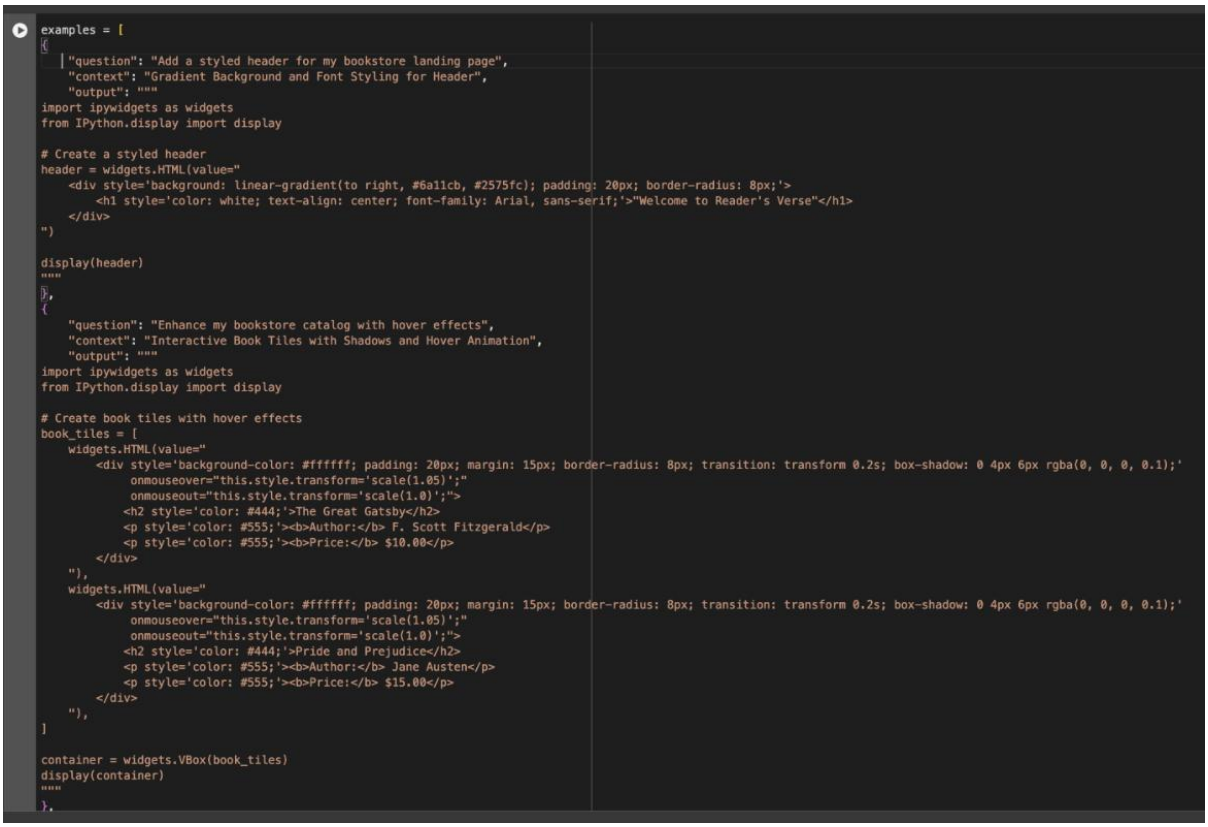
    Returns:
    - str, the generated result from the model
    """
    prompt = fewshot_prompt(context, question, book_titles, examples)
    result = model.invoke(prompt)

    return result
```

Catatan: Kamu tidak akan melihat output apa pun apabila kamu menjalankan sel ini di notebook.

3. Selanjutnya, kamu akan menyiapkan contoh elemen interaktif dengan pertanyaan, konteks, dan parameter output untuk memandu model IBM Granite dalam menghasilkan output yang sesuai dengan gaya dan format yang diminta oleh klien. Di skenario ini, contohnya termasuk format untuk gaya header, efek hover, dan elemen UI interaktif lainnya seperti yang diminta oleh klien.

   Pada sel kelima notebook, pilih tombol **Putar** untuk menyiapkan contoh yang akan digunakan model sambil memanggil fungsi **get_answer_using_fewshot** pada langkah berikutnya.



```
examples = [
{
    "question": "Add a styled header for my bookstore landing page",
    "context": "Gradient Background and Font Styling for Header",
    "output": """
import ipywidgets as widgets
from IPython.display import display

# Create a styled header
header = widgets.HTML(value="
    <div style='background: linear-gradient(to right, #6a11cb, #2575fc); padding: 20px; border-radius: 8px;'>
        <h1 style='color: white; text-align: center; font-family: Arial, sans-serif;'>"Welcome to Reader's Verse"</h1>
    </div>
")

display(header)
"""
},
{
    "question": "Enhance my bookstore catalog with hover effects",
    "context": "Interactive Book Tiles with Shadows and Hover Animation",
    "output": """
import ipywidgets as widgets
from IPython.display import display

# Create book tiles with hover effects
book_tiles = [
    widgets.HTML(value="
        <div style='background-color: #ffffff; padding: 20px; margin: 15px; border-radius: 8px; transition: transform 0.2s; box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);'
             onmouseover="this.style.transform='scale(1.05)';"
             onmouseout="this.style.transform='scale(1.0)';">
            <h2 style='color: #444;'>The Great Gatsby</h2>
            <p style='color: #555;'><b>Author:</b> F. Scott Fitzgerald</p>
            <p style='color: #555;'><b>Price:</b> $10.00</p>
        </div>
    "),
    widgets.HTML(value="
        <div style='background-color: #ffffff; padding: 20px; margin: 15px; border-radius: 8px; transition: transform 0.2s; box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);'
             onmouseover="this.style.transform='scale(1.05)';"
             onmouseout="this.style.transform='scale(1.0)';">
            <h2 style='color: #444;'>Pride and Prejudice</h2>
            <p style='color: #555;'><b>Author:</b> Jane Austen</p>
            <p style='color: #555;'><b>Price:</b> $15.00</p>
        </div>
    "),
]

container = widgets.VBox(book_tiles)
display(container)
"""
},
```

Catatan: Kamu tidak akan melihat output apa pun apabila kamu menjalankan sel ini di notebook.

4. Selanjutnya, kamu akan menjalankan fungsi **get_answer_using_fewshot** untuk menghasilkan output kode untuk halaman arahan Reader's Verse. Respons few-shot menyempurnakan output zero-shot dengan menggunakan contoh dan panduan kontekstual untuk menghasilkan kode untuk skenario yang diberikan. Prompt few-shot mencakup contoh yang telah disediakan sebelumnya yang menunjukkan cara menghasilkan kode Python untuk membuat UI interaktif untuk halaman arahan Reader's Verse.

   Pada sel keenam notebook, pilih tombol **Putar** untuk menjalankan fungsi **get_answer_using_fewshot**.

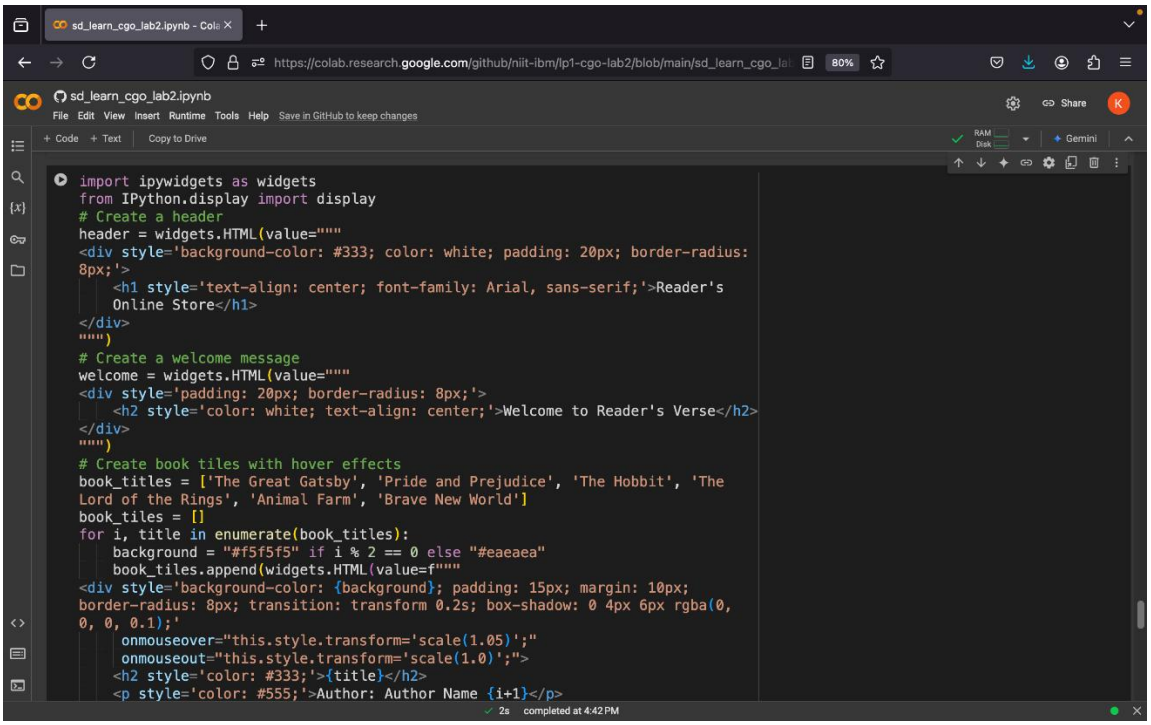
```
# Example usage
context = "Design and develop an online bookstore UI components with minimalistic theme."
question = "Create the landing page for users visiting my bookstore. The landing page should display a header `Reader's Online Store`
book_titles = ["The Great Gatsby", "Pride and Prejudice", "The Hobbit", "The Lord of the Rings", "Animal Farm", "Brave New World"]
# Generate and display the UI code for the landing page
result = get_answer_using_fewshot(context, question, book_titles, examples)
print(f"Generated Code:\n{result}")
```

5. Fungsi **get_answer_using_fewshot** menghasilkan dan menampilkan kode Python untuk UI halaman arahan mengikuti sel kode. Kode ini akan membuat halaman arahan termasuk judul, katalog judul buku, dan elemen UI interaktif.

   Catatan: Saat kamu membuat kode, mungkin sedikit berbeda dari output kode berikut. Perbedaan kecil ini adalah normal dan tidak akan memengaruhi fungsionalitas atau hasil lab secara keseluruhan.

```
Generated Code:
Here's the Python code for the requested task:
```python
import ipywidgets as widgets
from IPython.display import display
# Create a header
header = widgets.HTML(value="""
<div style='background-color: #333; color: white; padding: 20px; border-radius: 8px;'>
    <h1 style='text-align: center; font-family: Arial, sans-serif;'>Reader's Online Store</h1>
</div>
""")
# Create a welcome message
welcome = widgets.HTML(value="""
<div style='padding: 20px; border-radius: 8px;'>
    <h2 style='color: white; text-align: center;'>Welcome to Reader's Verse</h2>
</div>
""")
# Create book tiles
book_titles = ["The Great Gatsby", "Pride and Prejudice", "The Hobbit", "The Lord of the Rings", "Animal Farm", "Brave New World"]
book_tiles = []
for i, title in enumerate(book_titles):
    background = "#f5f5f5" if i % 2 == 0 else "#eaeaea"
    book_tiles.append(widgets.HTML(value=f"""
<div style='background-color: {background}; padding: 15px; margin: 10px; border-radius: 8px;'>
    <h2 style='color: #333;'>{title}</h2>
    <p style='color: #555;'>Author: Author Name {i+1}</p>
    <p style='color: #555;'>Price: ${(i+1)*10}.00</p>
</div>
"""))
# Create a container for the book tiles
book_container = widgets.VBox(book_tiles)
# Create a vertical box for the header, welcome message, and book container
main_container = widgets.VBox([header, welcome, book_container])
# Display the main container
display(main_container)
```
This code creates a landing page for an online bookstore with a header, welcome message, and a catalog of book titles. The book titles are displ
```

6. Untuk memudahkan eksekusi, kode output yang dihasilkan AI sudah terisi sebelumnya dalam sel baru Jupyter notebook. Pilih tombol **Putar** pada sel ketujuh pada notebook untuk memvalidasi output kode Python yang dihasilkan oleh model IBM Granite untuk skenario yang diberikan.



7. Outputnya akan menampilkan UI halaman arahan toko buku yang mengikuti sel kode, yang menampilkan komponen-komponen berikut:
    - Header bergaya
    - Daftar judul buku yang terformat dan terorganisir
    - Tema minimalis

Langkah 4: Mengoptimalkan kode yang dihasilkan AI

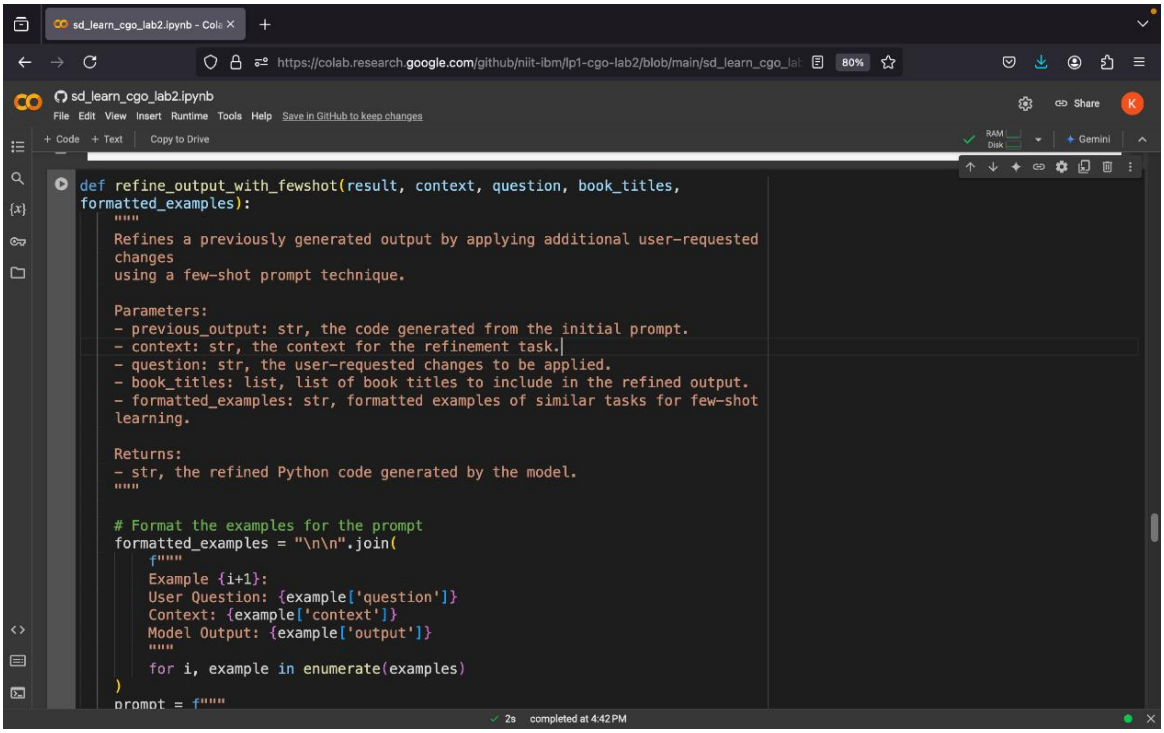
Ikhtisar

Dalam tugas ini, kamu akan mengoptimalkan output yang dihasilkan AI dari prompt few-shot dengan menambahkan elemen desain ke halaman arahan situs web Reader's Verse.

Instruksi

1. Menetapkan dan menjalankan fungsi yang menciptakan prompt few-shot untuk model dengan memberikan contoh untuk memandu model. Fungsi **refine_output_with_fewshot** menyempurnakan kode untuk meningkatkan pemformatan, gaya, dan fungsionalitas halaman arahan.

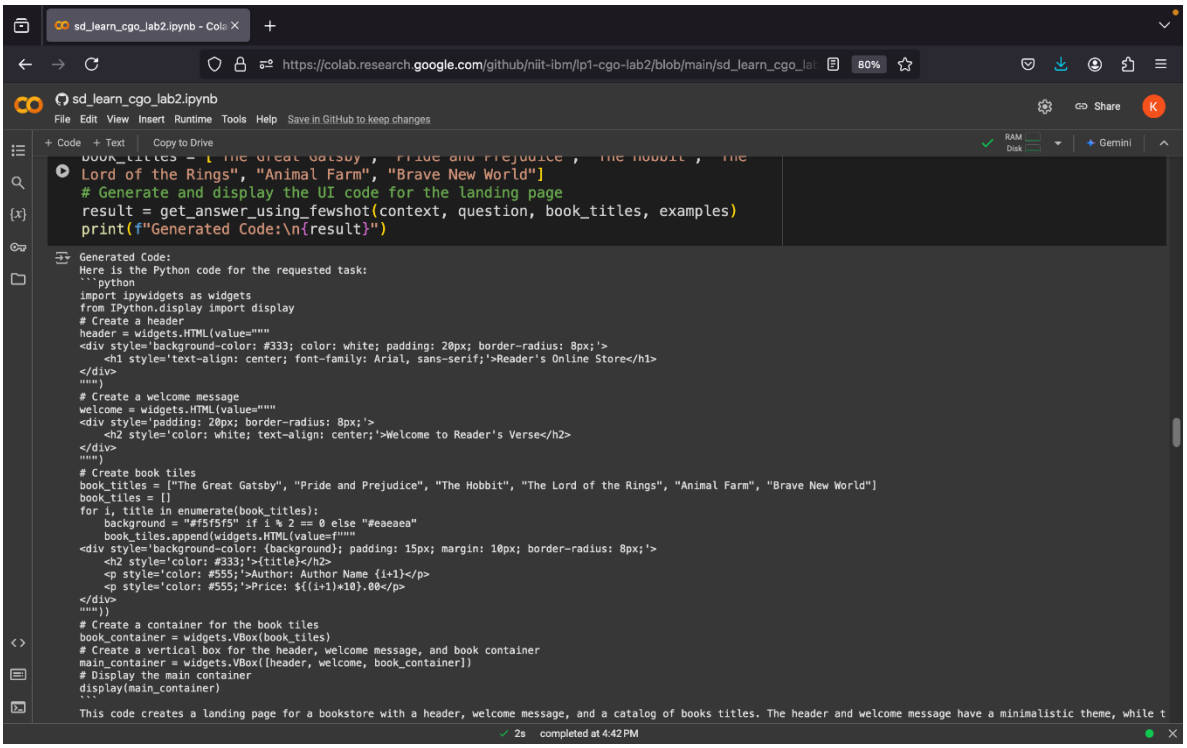
   Pada sel kedelapan notebook, pilih tombol **Putar** untuk menentukan fungsi **refine_output_with_fewshot** yang berisi serangkaian instruksi yang memandu model pada output yang akan dihasilkan. Dalam skenario ini, instruksi-instruksi tersebut menginformasikan model untuk menyempurnakan kode untuk halaman arahan berdasarkan parameter yang diberikan: output dari prompt few-shot, konteks, pertanyaan, judul buku, dan contoh-contoh yang telah diformat.



2. Pada sel kesembilan notebook, pilih tombol **Putar** untuk menjalankan fungsi **refine_output_with_fewshot.** Fungsi ini menyempurnakan output awal dari prompt few-shot untuk menghasilkan kode Python yang menambahkan efek bayangan dan hover untuk menyempurnakan desain halaman arahan untuk situs web Reader's Verse.

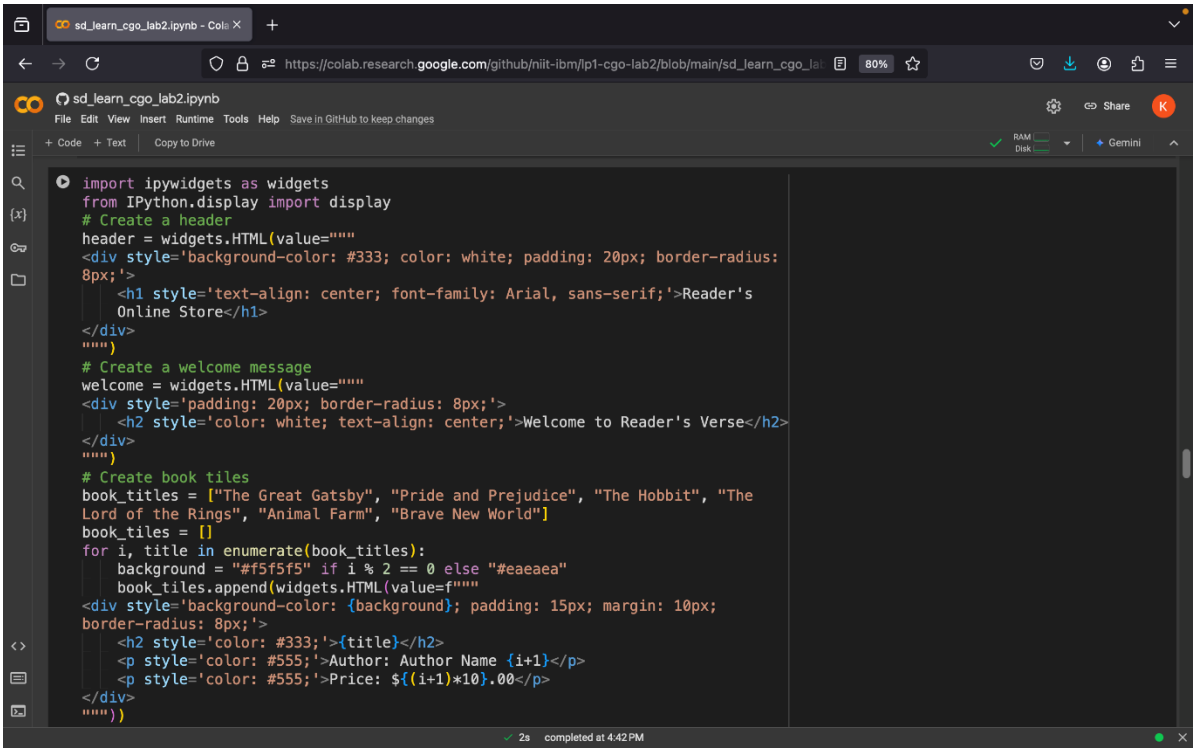
```
# Generate the Code
context = "Refined page with Hover Effects for Enhanced User Interaction"
question = "Enhance the book tiles in the catalog with hover effects that
change the background color, add a shadow, and slightly scale the tiles on
hover. Output should also retain the Header and Welcome message aswell in the
page"
# Generate and display the UI code for the landing page
ref_result = refine_output_with_fewshot(context, question, book_titles,
examples, result)
print(f"Generated Code:\n{ref_result}")
```

3. Output yang dihasilkan mencakup kode Python yang telah disempurnakan yang akan membuat UI halaman arahan toko buku, termasuk hak milik dan katalog judul buku dengan peningkatan rancang yang diminta oleh klien.

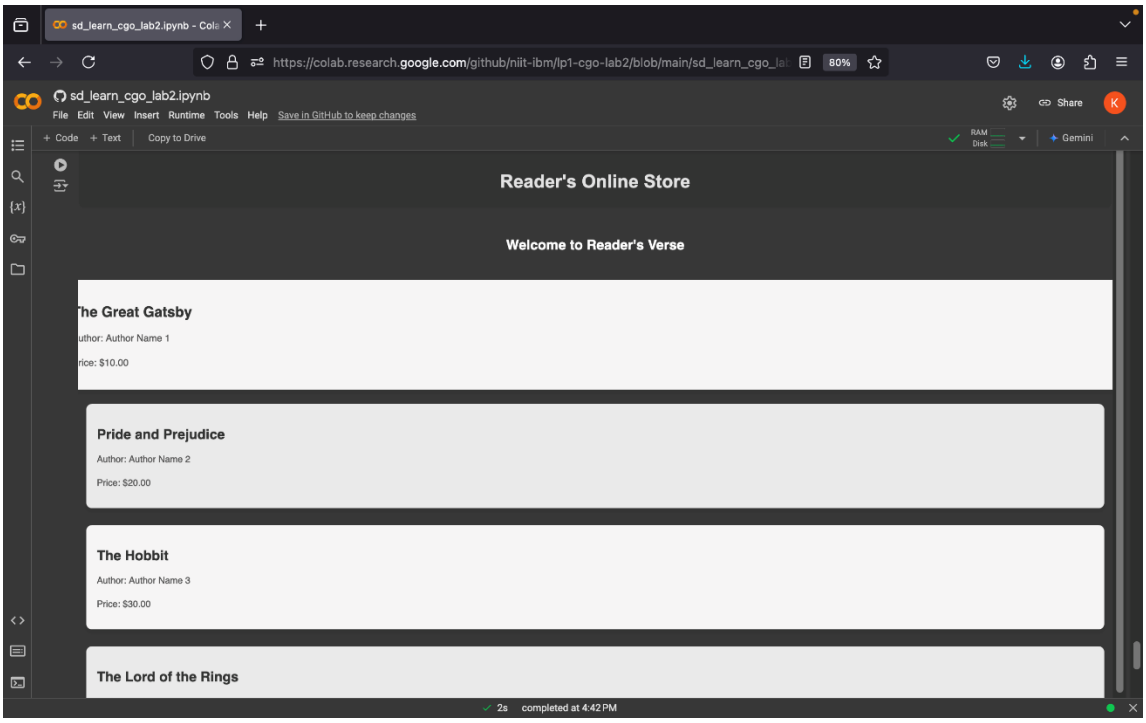


Catatan: Saat kamu membuat kode, mungkin sedikit berbeda dari output kode berikut. Perbedaan kecil ini adalah normal dan tidak akan memengaruhi fungsionalitas atau hasil lab secara keseluruhan.

4. Untuk memudahkan eksekusi, kode output sudah diisi sebelumnya dalam sel baru pada Jupyter notebook. Pilih tombol **Putar** pada sel kesepuluh notebook untuk memvalidasi output kode Python yang telah disempurnakan yang dihasilkan oleh model IBM Granite untuk skenario yang diberikan.



5. Outputnya akan menampilkan UI halaman arahan toko buku yang mengikuti sel kode. Kamu akan mengamati penyempurnaan berikut ini dalam desain halaman arahan dibandingkan dengan hasil awal prompt few-shot:
    - Efek melayang untuk ubin buku
    - Efek bayangan dan transisi yang mulus
    - Pemformatan yang konsisten



Kesimpulan

Kamu menggunakan model IBM Granite untuk mengoptimalkan kode untuk membuat halaman arahan situs web untuk Reader's Verse, sebuah toko buku online hipotetis. Klien senang melihat betapa cepatnya kamu membuat prototipe UI yang berfungsi untuk halaman arahan situs web. Kamu menggunakan prompt few-shot untuk menghasilkan output kode Python dari model IBM Granite dan memvalidasi kode yang dihasilkan AI untuk memastikan bahwa kode tersebut merender output yang diharapkan dengan benar.