CAPSTONE PROJECT

AI AGENT FOR SMART FARMING ADVICE

Presented By:
Student Name- DILIP RAHUL BUDIMURI
College Name- NIMRA COLLEGE OF ENGINEERING AND TECHNOLOGY
Department- COMPUTER SCIENCE AND ENGINEERING

OUTLINE

- Problem Statement
- Technologies and Services used
- Proposed Solution
- WOW Factor
- Algorithm & Deployment
- Overview & Result
- Conclusion
- References
- Future Scope
- IBM Certifications
- GitHub Link

PROBLEM STATEMENT

- Small-scale farmers often lack access to timely and localized agricultural advice. Unpredictable weather, poor soil management, pest attacks, and fluctuating crop prices lead to reduced yield and income.
- The challenge is to build an AI agent using Retrieval-Augmented Generation (RAG) that provides real-time, region-specific guidance on crops, weather, soil, pest control, and market prices — accessible in local languages — to empower farmers with data-driven decisions.

TECHNOLOGIES AND SERVICES USED

- IBM Granite (language generation)
- IBM Cloud Lite (retrieval and hosting)
- Vector Database
- Presto for querying structured Agri-data
- Python, LangChain (for orchestration)
- REST API & RAG workflows

PROPOSED SOLUTION

- Overview of the solution: An AI agent built on IBM Cloud Lite and Granite, using retrieval and language generation to answer farming queries.
  Key features:
- Crop suggestions adapted to weather and region
- Fertilizer advice and soil analysis
- Pest management tips
- Real-time market price info
  Emphasizes integration of trusted sources, ensuring reliable and practical advice.

WOW FACTORS

1. **Real-Time, Contextual Advice via RAG**

- Retrieves latest, region-specific data for each query
- Generates advice that adapts to weather, soil, & local trends

2. **End-to-End IBM Cloud Solution**

- Fully hosted on IBM Cloud Lite, no third-party tools needed
- Seamlessly connects storage, functions, and AI services in one platform

3. **Decision-Making, Not Just Information**

- Converts raw data into clear, actionable recommendations
- Provides step-by-step guidance, boosting farmer confidence

ALGORITHM & DEPLOYMENT

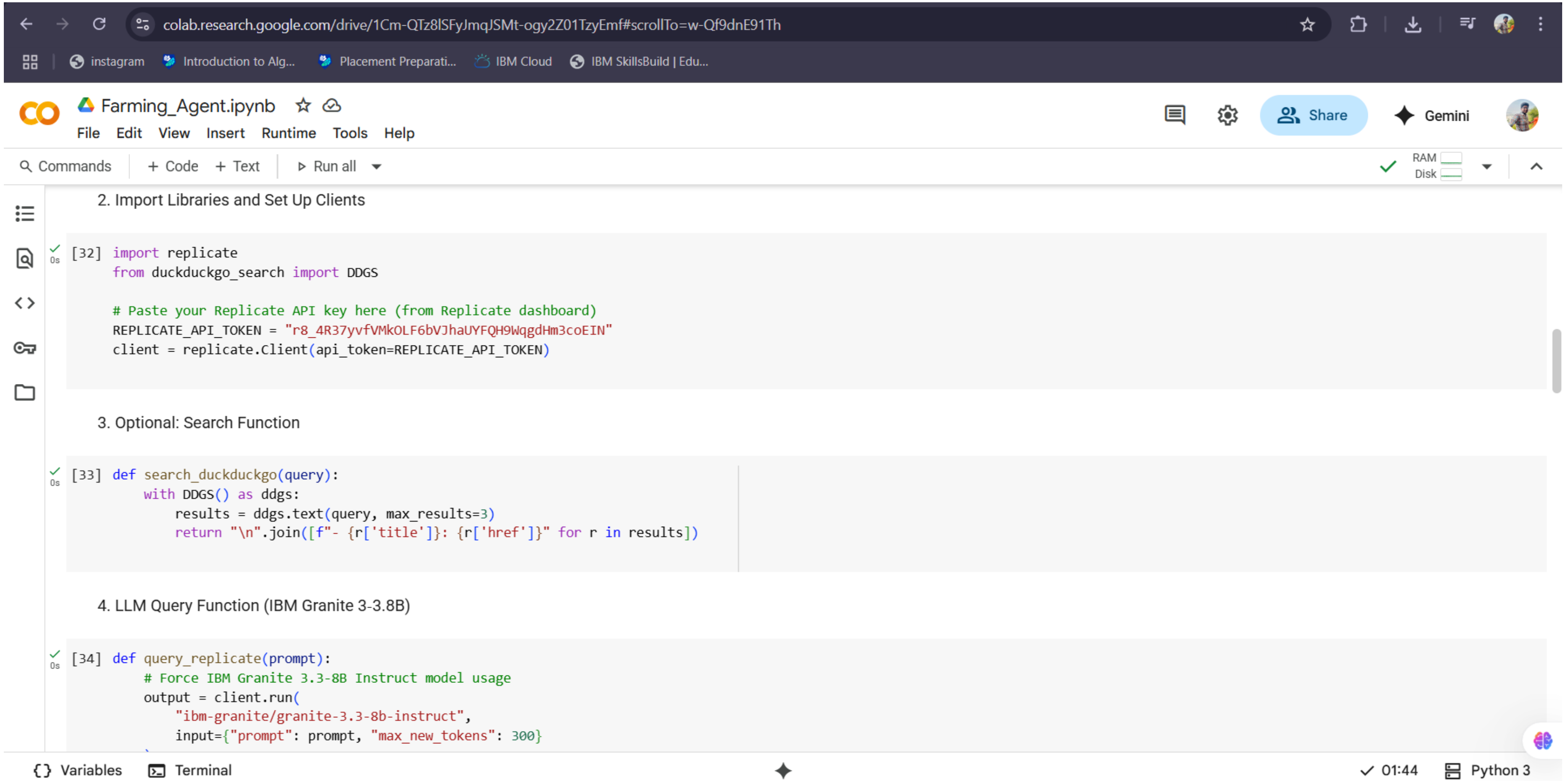
Algorithm Selection:

- Retrieval-Augmented Generation (RAG) + Embedding model (e.g., BAAI/bge)
- Vector index enables semantic search of soil, weather, pest, and mandi data
- IBM Granite model is used for generating the final answer

Deployment Steps:

- Create cloud functions for retrieval
- Hosted frontend on Gradio
- Deploy agent via web UI / chatbot

OVERVIEW



2. Import Libraries and Set Up Clients

```
[32] import replicate
     from duckduckgo_search import DDGS

     # Paste your Replicate API key here (from Replicate dashboard)
     REPLICATE_API_TOKEN = "r8_4R37yvfVMkOLF6bVJhaUYFQH9WqgdHm3coEIN"
     client = replicate.Client(api_token=REPLICATE_API_TOKEN)
```

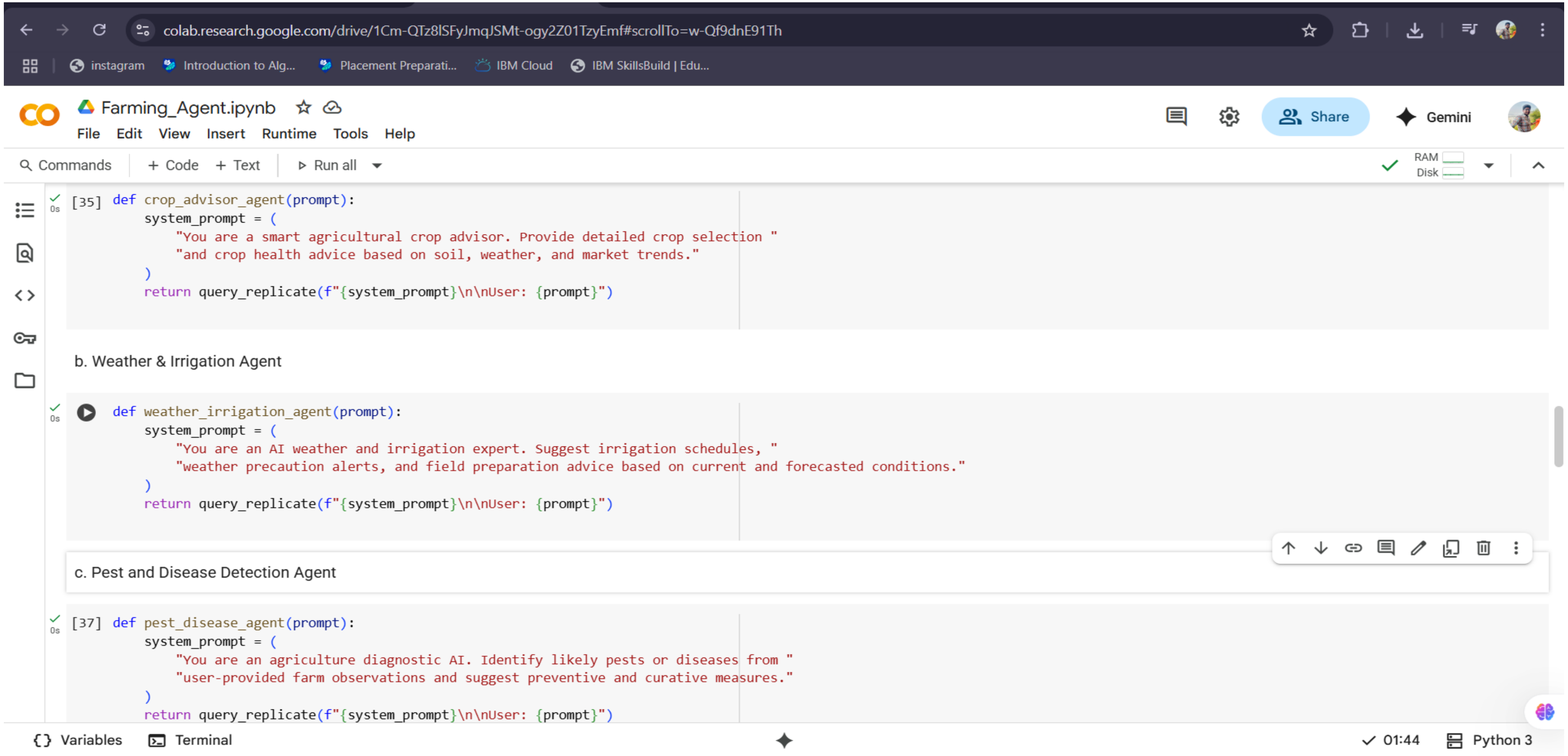
3. Optional: Search Function

```
[33] def search_duckduckgo(query):
         with DDGS() as ddgs:
             results = ddgs.text(query, max_results=3)
             return "\n".join([f"- {r['title']}: {r['href']}" for r in results])
```

4. LLM Query Function (IBM Granite 3-3.8B)

```
[34] def query_replicate(prompt):
         # Force IBM Granite 3.3-8B Instruct model usage
         output = client.run(
             "ibm-granite/granite-3.3-8b-instruct",
             input={"prompt": prompt, "max_new_tokens": 300}
```

OVERVIEW



```
[35] def crop_advisor_agent(prompt):
         system_prompt = (
             "You are a smart agricultural crop advisor. Provide detailed crop selection "
             "and crop health advice based on soil, weather, and market trends."
         )
         return query_replicate(f"{system_prompt}\n\nUser: {prompt}")
```

b. Weather & Irrigation Agent

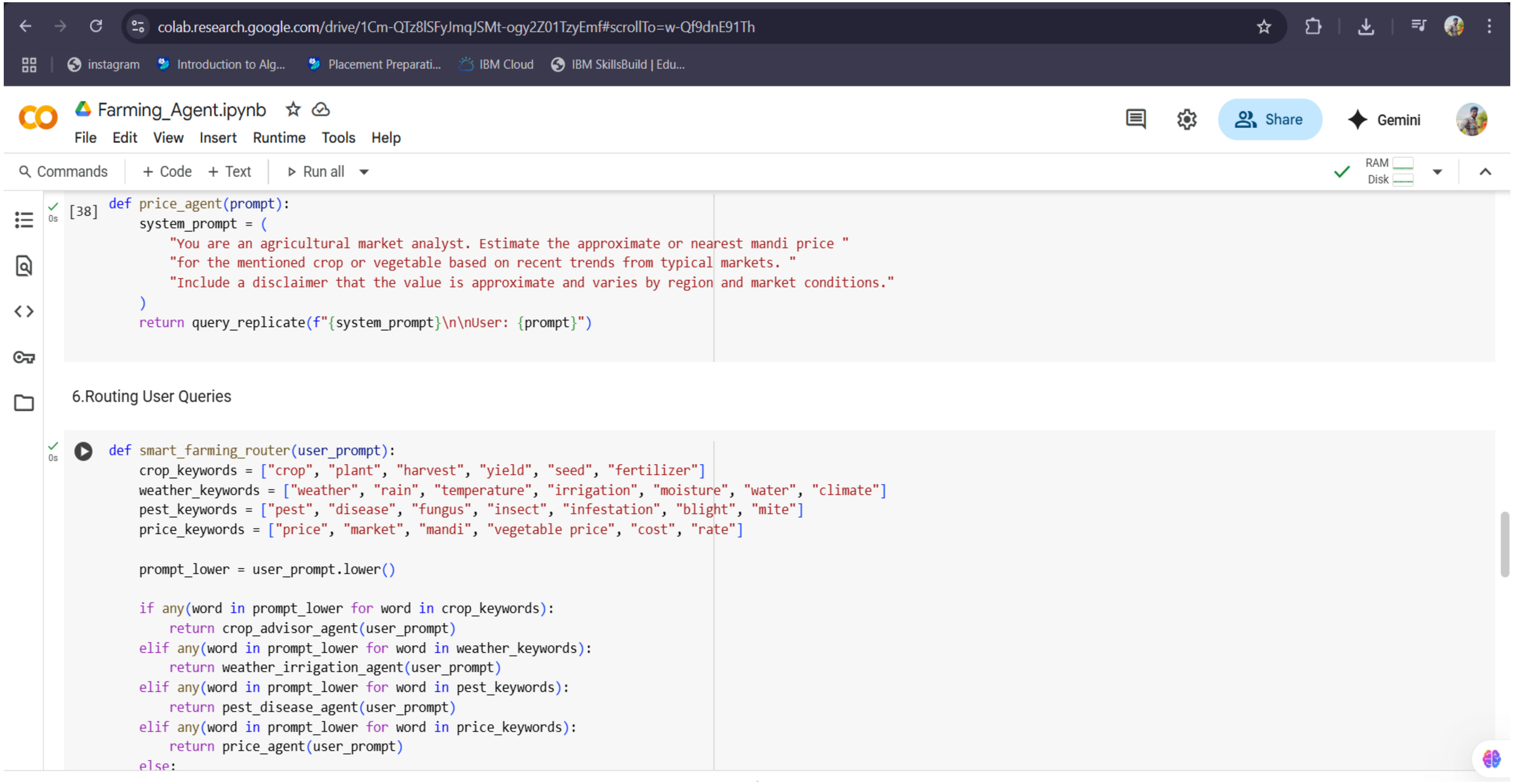
```
def weather_irrigation_agent(prompt):
    system_prompt = (
        "You are an AI weather and irrigation expert. Suggest irrigation schedules, "
        "weather precaution alerts, and field preparation advice based on current and forecasted conditions."
    )
    return query_replicate(f"{system_prompt}\n\nUser: {prompt}")
```

c. Pest and Disease Detection Agent

```
[37] def pest_disease_agent(prompt):
         system_prompt = (
             "You are an agriculture diagnostic AI. Identify likely pests or diseases from "
             "user-provided farm observations and suggest preventive and curative measures."
         )
         return query_replicate(f"{system_prompt}\n\nUser: {prompt}")
```

edunet
foundation

OVERVIEW

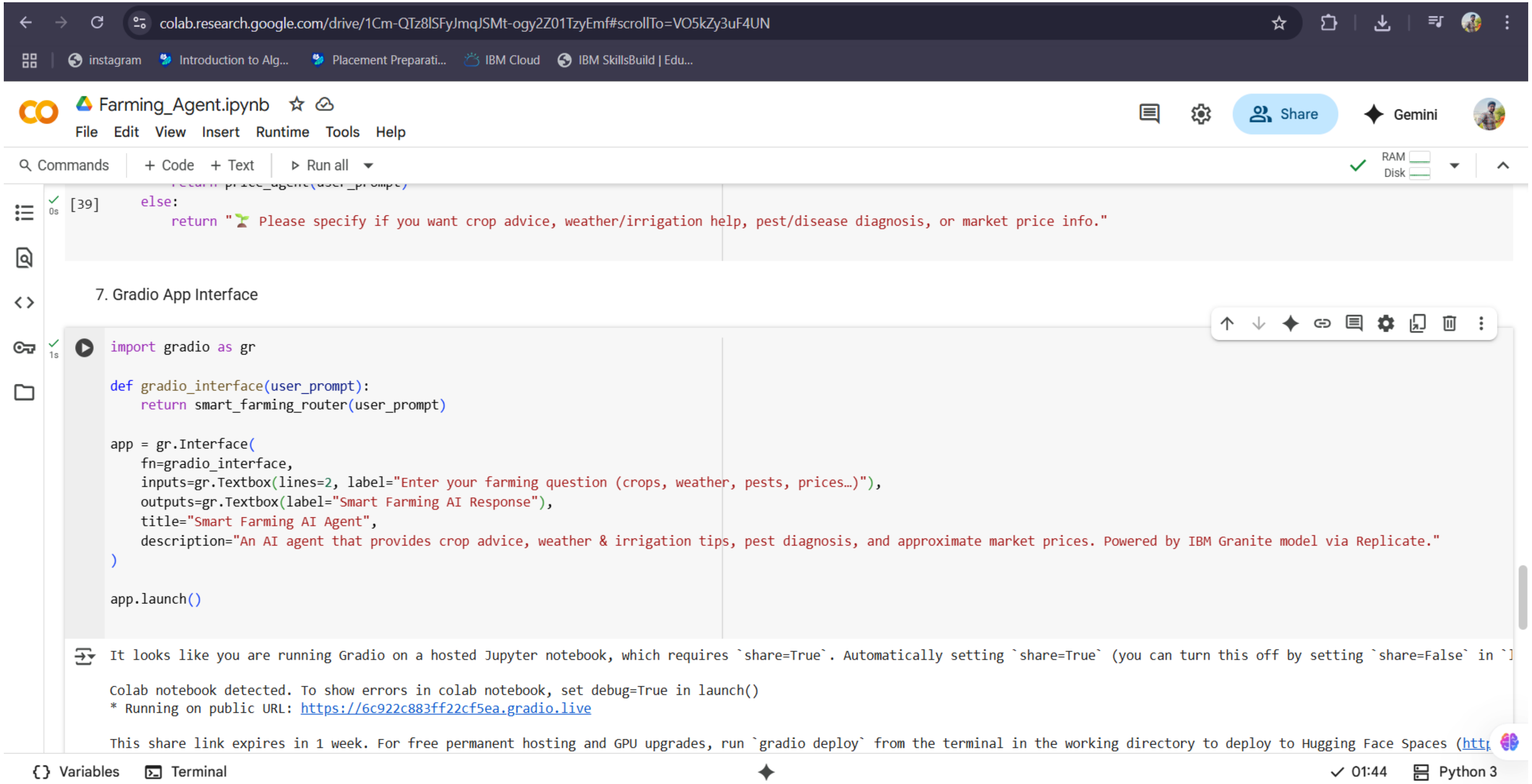


[38]
```python
def price_agent(prompt):
    system_prompt = (
        "You are an agricultural market analyst. Estimate the approximate or nearest mandi price "
        "for the mentioned crop or vegetable based on recent trends from typical markets. "
        "Include a disclaimer that the value is approximate and varies by region and market conditions."
    )
    return query_replicate(f"{system_prompt}\n\nUser: {prompt}")
```

6.Routing User Queries

```python
def smart_farming_router(user_prompt):
    crop_keywords = ["crop", "plant", "harvest", "yield", "seed", "fertilizer"]
    weather_keywords = ["weather", "rain", "temperature", "irrigation", "moisture", "water", "climate"]
    pest_keywords = ["pest", "disease", "fungus", "insect", "infestation", "blight", "mite"]
    price_keywords = ["price", "market", "mandi", "vegetable price", "cost", "rate"]

    prompt_lower = user_prompt.lower()

    if any(word in prompt_lower for word in crop_keywords):
        return crop_advisor_agent(user_prompt)
    elif any(word in prompt_lower for word in weather_keywords):
        return weather_irrigation_agent(user_prompt)
    elif any(word in prompt_lower for word in pest_keywords):
        return pest_disease_agent(user_prompt)
    elif any(word in prompt_lower for word in price_keywords):
        return price_agent(user_prompt)
    else:
```

OVERVIEW



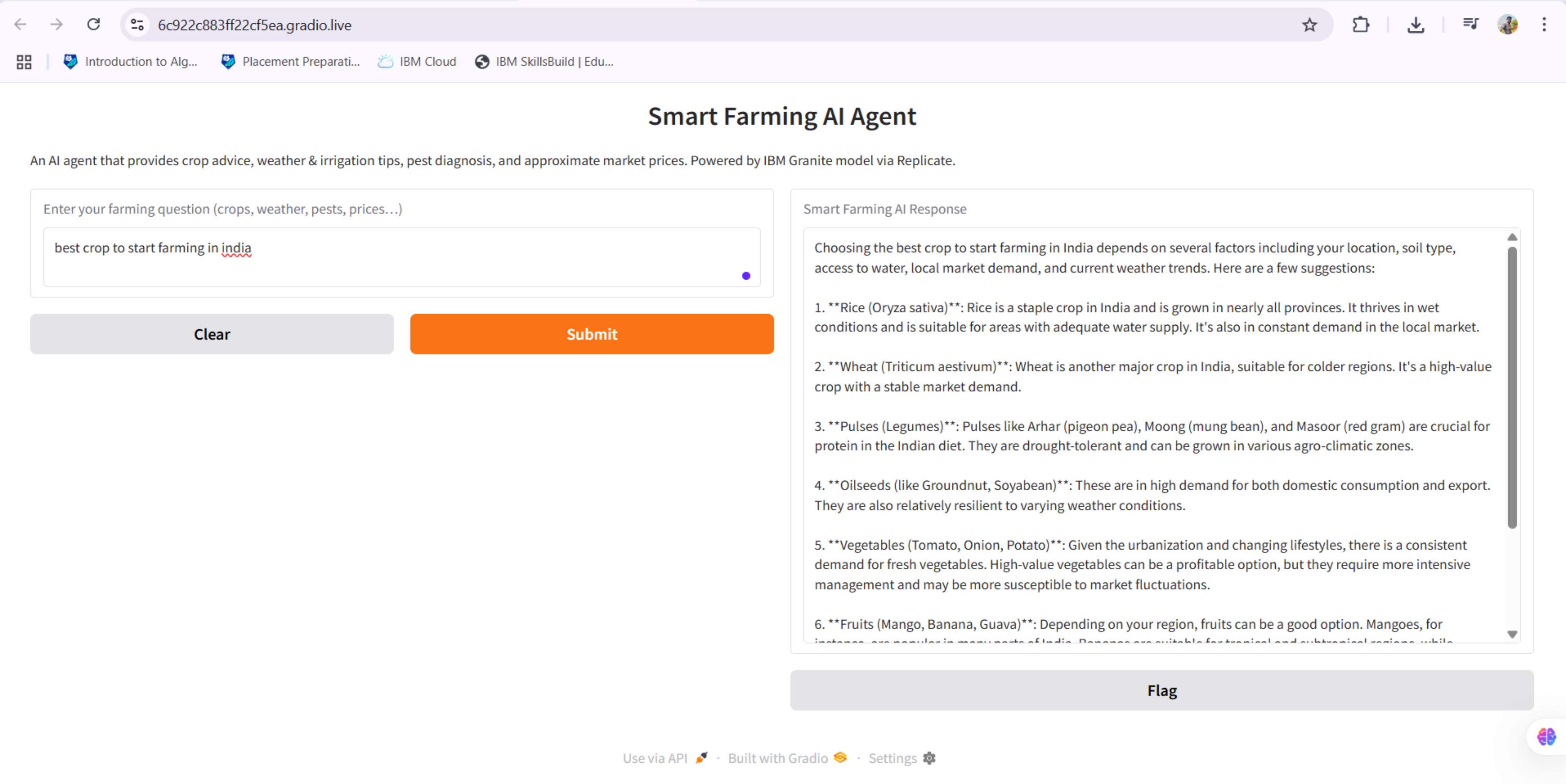
```
[39]    else:
            return "🌱 Please specify if you want crop advice, weather/irrigation help, pest/disease diagnosis, or market price info."
```

7. Gradio App Interface

```
import gradio as gr

def gradio_interface(user_prompt):
    return smart_farming_router(user_prompt)

app = gr.Interface(
    fn=gradio_interface,
    inputs=gr.Textbox(lines=2, label="Enter your farming question (crops, weather, pests, prices…)"),
    outputs=gr.Textbox(label="Smart Farming AI Response"),
    title="Smart Farming AI Agent",
    description="An AI agent that provides crop advice, weather & irrigation tips, pest diagnosis, and approximate market prices. Powered by IBM Granite model via Replicate."
)

app.launch()
```

```
It looks like you are running Gradio on a hosted Jupyter notebook, which requires `share=True`. Automatically setting `share=True` (you can turn this off by setting `share=False` in `l

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://6c922c883ff22cf5ea.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging Face Spaces (http
```

RESULT



Smart Farming AI Agent

An AI agent that provides crop advice, weather & irrigation tips, pest diagnosis, and approximate market prices. Powered by IBM Granite model via Replicate.

Enter your farming question (crops, weather, pests, prices…)

best crop to start farming in india

Smart Farming AI Response

Choosing the best crop to start farming in India depends on several factors including your location, soil type, access to water, local market demand, and current weather trends. Here are a few suggestions:

1. **Rice (Oryza sativa)**: Rice is a staple crop in India and is grown in nearly all provinces. It thrives in wet conditions and is suitable for areas with adequate water supply. It's also in constant demand in the local market.

2. **Wheat (Triticum aestivum)**: Wheat is another major crop in India, suitable for colder regions. It's a high-value crop with a stable market demand.

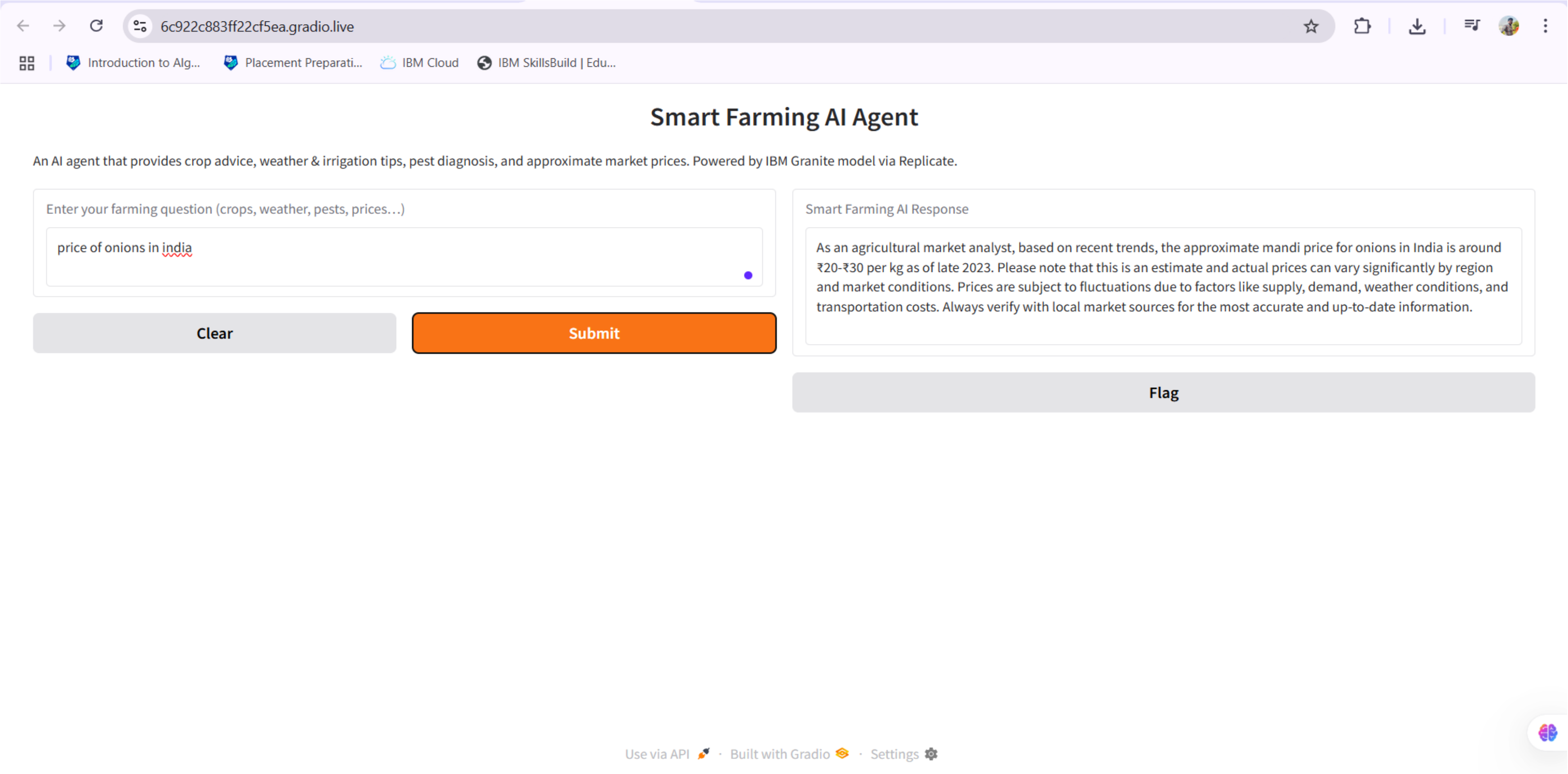
3. **Pulses (Legumes)**: Pulses like Arhar (pigeon pea), Moong (mung bean), and Masoor (red gram) are crucial for protein in the Indian diet. They are drought-tolerant and can be grown in various agro-climatic zones.

4. **Oilseeds (like Groundnut, Soyabean)**: These are in high demand for both domestic consumption and export. They are also relatively resilient to varying weather conditions.

5. **Vegetables (Tomato, Onion, Potato)**: Given the urbanization and changing lifestyles, there is a consistent demand for fresh vegetables. High-value vegetables can be a profitable option, but they require more intensive management and may be more susceptible to market fluctuations.

6. **Fruits (Mango, Banana, Guava)**: Depending on your region, fruits can be a good option. Mangoes, for instance, are popular in many parts of India. Bananas are suitable for tropical and subtropical regions, while

RESULT



Smart Farming AI Agent

An AI agent that provides crop advice, weather & irrigation tips, pest diagnosis, and approximate market prices. Powered by IBM Granite model via Replicate.

Enter your farming question (crops, weather, pests, prices…)

price of onions in india

Smart Farming AI Response

As an agricultural market analyst, based on recent trends, the approximate mandi price for onions in India is around ₹20-₹30 per kg as of late 2023. Please note that this is an estimate and actual prices can vary significantly by region and market conditions. Prices are subject to fluctuations due to factors like supply, demand, weather conditions, and transportation costs. Always verify with local market sources for the most accurate and up-to-date information.

CONCLUSION

- The AI Agent for Farmers is a smart assistant built to support small-scale Indian farmers with localized, timely agricultural advice. Powered by IBM Cloud Lite and IBM Granite using Retrieval-Augmented Generation (RAG), it helps with crop guidance, soil health, weather forecasts, pest control, and real-time mandi prices. It retrieves trusted data from sources like ensuring practical and region-specific responses.
- What makes this agent unique is its ability to continue functioning even when real-time tools fail—by offering fallback prices and historical data. It’s multilingual, simple to use, and provides both technical and economic farming help. Overall, it acts as a reliable digital guide from sowing to market, making farming more informed and efficient.

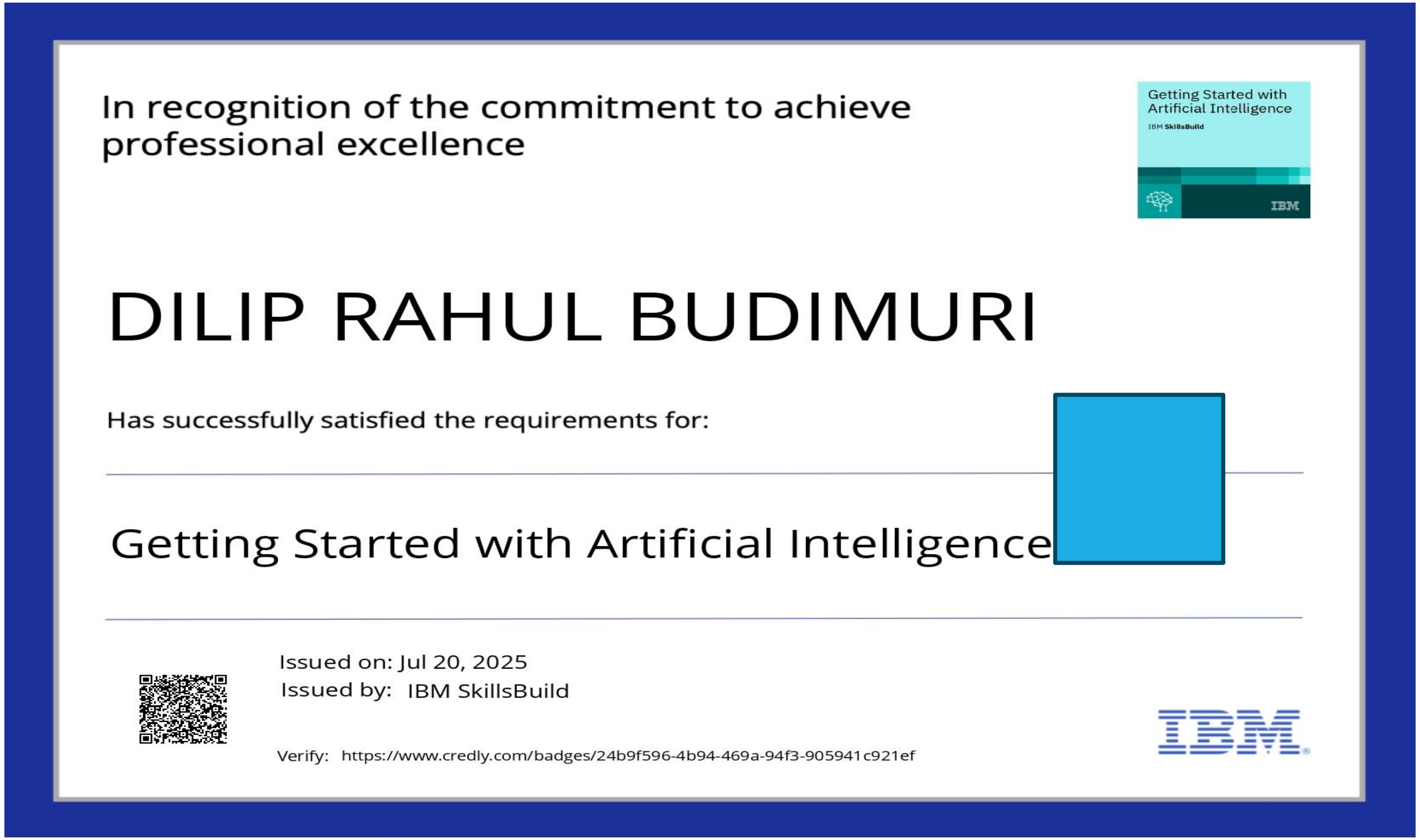
REFERENCES

•**IBM Cloud Lite** – Used for deploying and managing backend services.
https://www.ibm.com/cloud/free
•**IBM Granite Foundation Models** – Used for natural language understanding and generation.
https://www.ibm.com/products/granite

FUTURE SCOPE

- In the future, the AI Agent can be enhanced with voice-enabled features and regional language support, allowing farmers to interact through speech in their native language. Integrating satellite data and advanced image recognition could help with real-time crop disease detection and soil health monitoring using photos or live drone feeds.
- Additionally, real-time integration with mandi APIs, IoT sensors, and mobile USSD access can make the agent usable even in low-internet or remote areas. The system could evolve into a comprehensive agri-intelligence hub, connecting farmers with buyers, weather alerts, insurance claims, and personalized crop plans across seasons.

IBM CERTIFICATIONS



Getting Started with Artificial Intelligence

IBM SkillsBuild

In recognition of the commitment to achieve professional excellence

DILIP RAHUL BUDIMURI

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

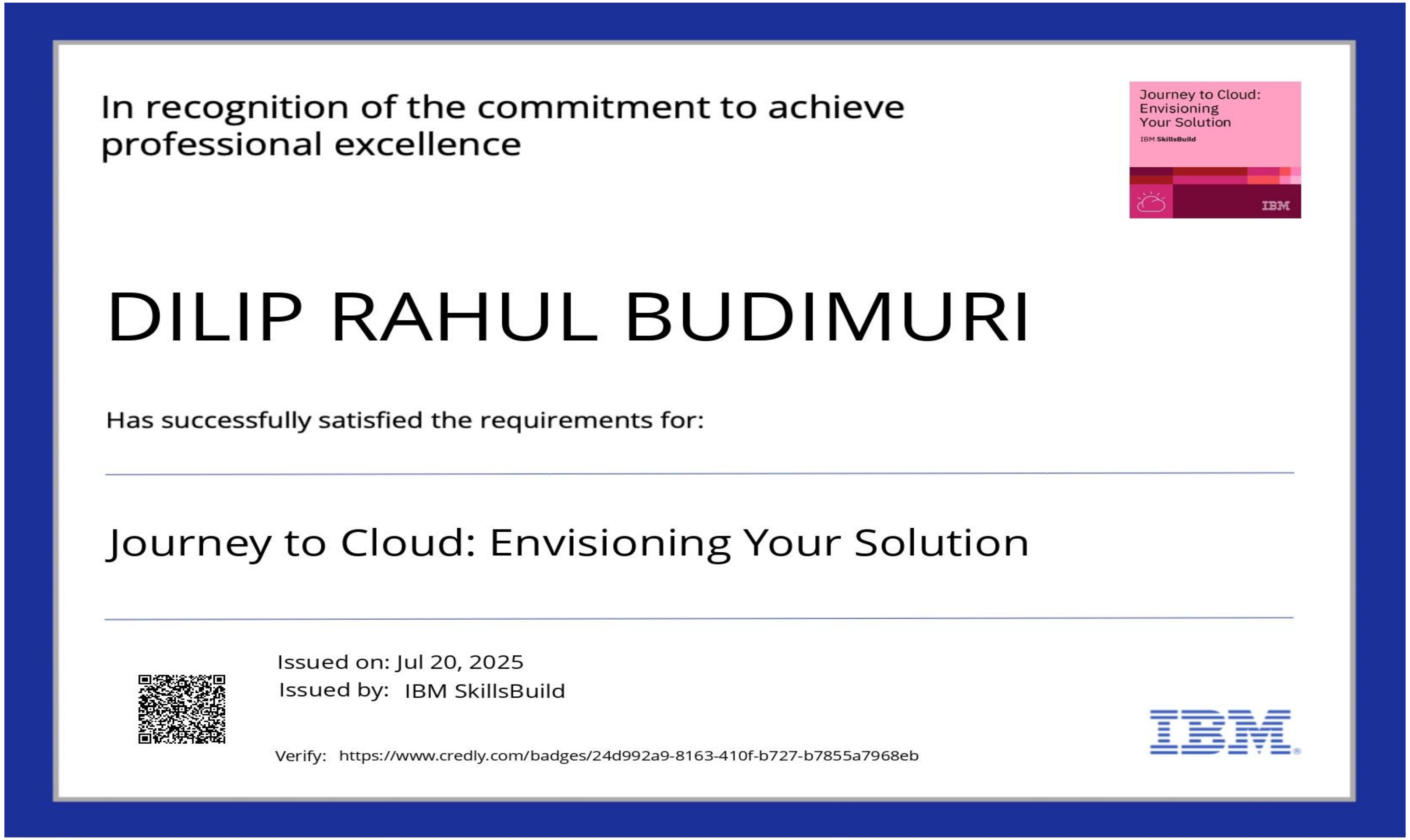
Issued on: Jul 20, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/24b9f596-4b94-469a-94f3-905941c921ef

IBM

IBM CERTIFICATIONS



Journey to Cloud:
Envisioning
Your Solution
IBM SkillsBuild

In recognition of the commitment to achieve professional excellence

DILIP RAHUL BUDIMURI

Has successfully satisfied the requirements for:

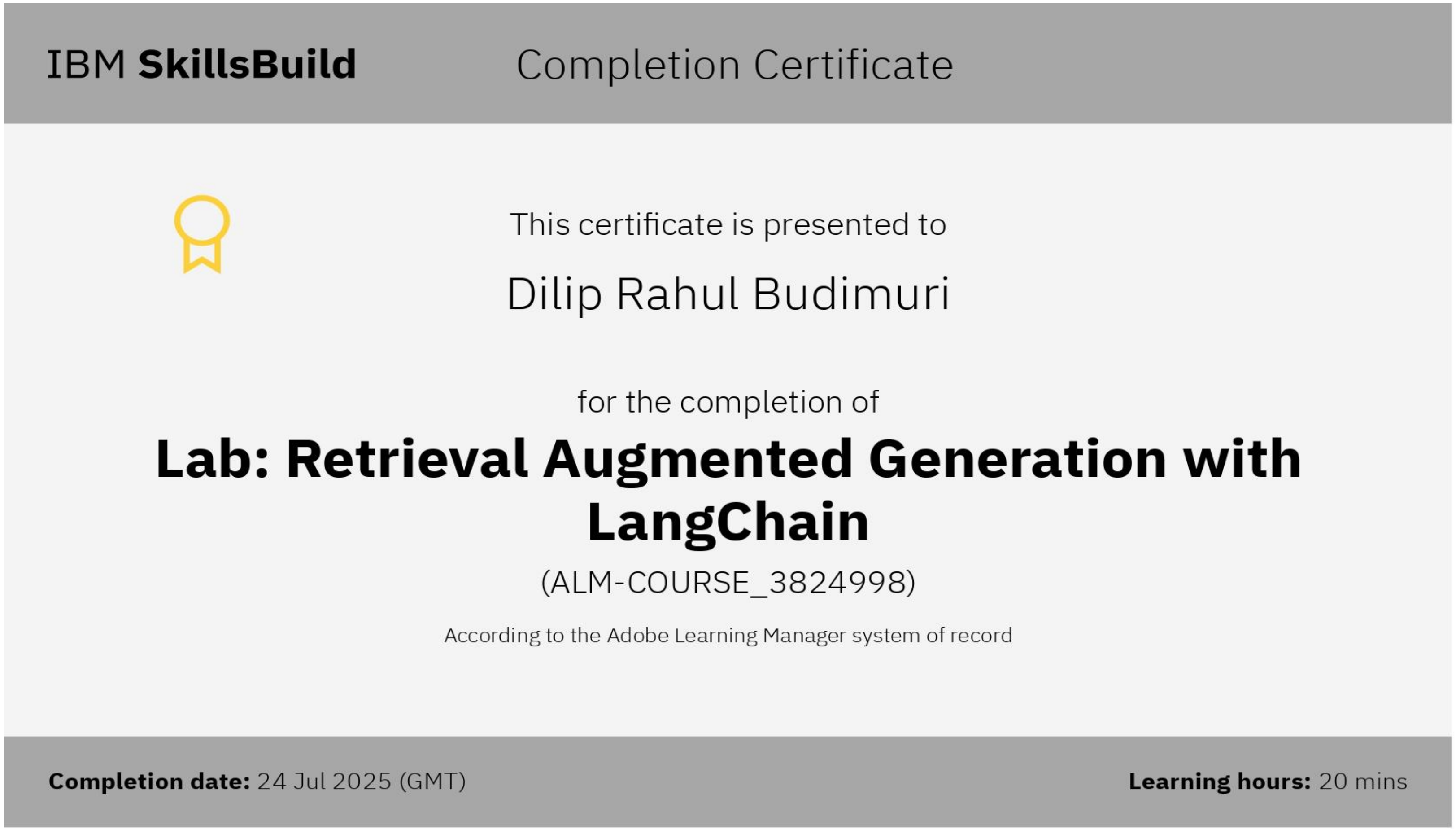
Journey to Cloud: Envisioning Your Solution

Issued on: Jul 20, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/24d992a9-8163-410f-b727-b7855a7968eb

IBM

IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Dilip Rahul Budimuri

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 24 Jul 2025 (GMT)

Learning hours: 20 mins

GITHUB LINK

GitHub link :- https://github.com/diliprahul/AI_AGENT_FOR_FARMERS

THANK YOU