Python for Accounting Applications

Applications of Accounting Data Analytics with Python

1131PAA11
ACC2, NTPU (U2004) (Fall 2024)
Wed 6, 7, 8, (14:10-17:00) (9:10-12:00) (B3F10)

Min-Yuh Day, Ph.D,
Professor

Institute of Information Management, National Taipei University

https://web.ntpu.edu.tw/~myday

2024-12-11

Syllabus

Week	Date	Subject/Topics
1	2024/09/11	Introduction to Python for Accounting Applications
2	2024/09/18	Python Programming and Data Science
3	2024/09/25	Foundations of Python Programming
4	2024/10/02	Data Structures
5	2024/10/09	Control Logic and Loops
6	2024/10/16	Functions and Modules; Files and Exception Handling
7	2024/10/23	Data Analytics and Visualization with Python
8	2024/10/30	~~Midterm Project Report~~ (Self-Learning)

Syllabus

Week	Date	Subject/Topics
9	2024/11/06	Self-Learning
10	2024/11/13	Midterm Project Report
11	2024/11/20	Obtaining Data From the Web with Python; Statistical Analysis with Python
12	2024/11/27	Machine Learning with Python
13	2024/12/04	Text Analytics with Generative AI and Python
14	2024/12/11	Applications of Accounting Data Analytics with Python
15	2024/12/18	Applications of ESG Data Analytics with Python
16	2024/12/25	Final Project Report

Applications of Accounting Data Analytics with Python

https://colab.research.google.com/drive/1IZnqcenbxSlehvPYb3xMqLVF21oWQdj_?usp=sharing

Audit Data Classification

https://www.kaggle.com/datasets/sid321axn/audit-data/data

kaggle

Search

Create

Home

Competitions

Datasets

Models

Code

Discussions

Learn

More

Your Work

VIEWED

EDITED

View Active Events

MANU SIDDHARTHA · UPDATED 5 YEARS AGO

86 | New Notebook | Download

Audit Data

Audit Risk dataset for classifying fraudulent firms

Data Card | Code (19) | Discussion (2) | Suggestions (0)

About Dataset

Context

Audit Risk Dataset for classifying Fraudulent Firms

Content

The goal of the dataset is to help the auditors by building a classification model that can predict the fraudulent firm on the basis the present and historical risk factors. The information about the sectors and the counts of firms are listed respectively as Irrigation (114), Public Health (77), Buildings and Roads (82), Forest (70), Corporate (47), Animal Husbandry (95), Communication (1), Electrical (4), Land (5), Science and Technology (3), Tourism (1), Fisheries (41), Industries (37), Agriculture (200).

This research work is a case study of an external government audit company which is also the external auditor of government firms of India. During audit-planning, auditors examine the business of different government offices but the target to visit the offices with very-high likelihood and significance of misstatements. This is calculated by assessing the

Usability

8.53

License

CC0: Public Domain

Expected update frequency

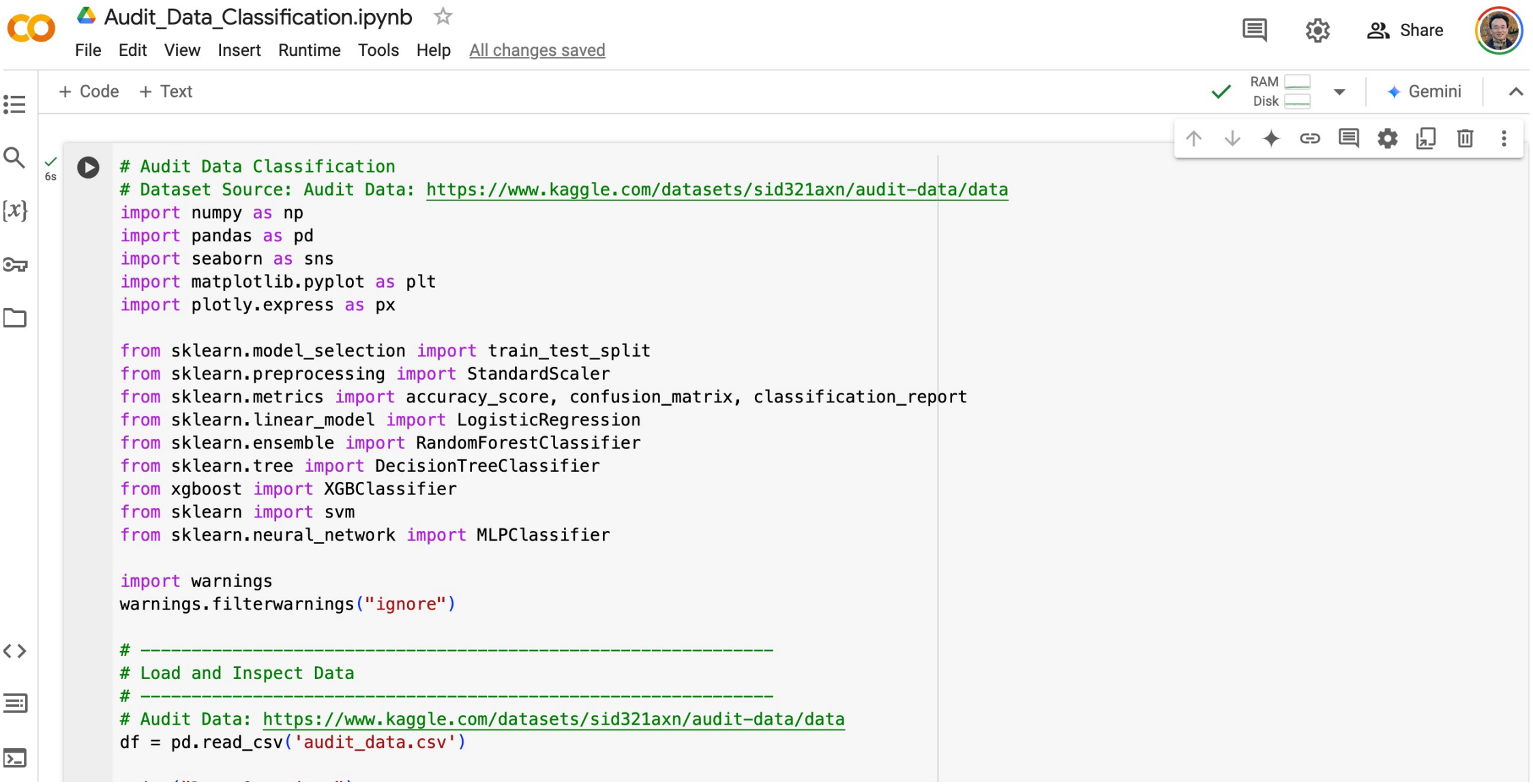
Not specified

Tags

Business | Software | Finance | Classification | Binary Classification

Audit Data Classification

https://www.kaggle.com/datasets/sid321axn/audit-data/data



https://colab.research.google.com/drive/1IZnqcenbxSlehvPYb3xMqLVF21oWQdj_?usp=sharing

```
# Audit Data Classification
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
from sklearn import svm
from sklearn.neural_network import MLPClassifier

import warnings
warnings.filterwarnings("ignore")
```

https://tinyurl.com/aintpupython101

```
# Audit Data: https://www.kaggle.com/datasets/sid321axn/audit-data/data
# Load and Inspect Data
df = pd.read_csv('audit_data.csv')

print("Data Overview:")
print(df.head(), "\n")
print("Missing Values:")
print(df.isnull().sum(), "\n")

# Visualize target distribution
sns.countplot(x="Risk", data=df, palette="coolwarm")
plt.title("Count of Risk Labels")
plt.show()
```

https://tinyurl.com/aintpupython101

```
# Data Preprocessing
df_ml = df.copy()

# Fill missing values in 'Money_Value' with the mean
df_ml['Money_Value'] =
df_ml['Money_Value'].fillna(df_ml['Money_Value'].mean())

# One-hot encode the 'LOCATION_ID' column
location_dummies = pd.get_dummies(df_ml['LOCATION_ID'],
prefix='location')
df_ml = pd.concat([df_ml, location_dummies],
axis=1).drop('LOCATION_ID', axis=1)
```

https://tinyurl.com/aintpupython101

```
# Separate features and target
y = df_ml['Risk']
X = df_ml.drop('Risk', axis=1)

# Split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(
X, y, train_size=0.7, shuffle=True, random_state=1
)
```

https://tinyurl.com/aintpupython101

```
# Standardize the numerical features
scaler = StandardScaler()
scaler.fit(X_train)
X_train = pd.DataFrame(scaler.transform(X_train),
index=X_train.index, columns=X_train.columns)
X_test = pd.DataFrame(scaler.transform(X_test),
index=X_test.index, columns=X_test.columns)
```

https://tinyurl.com/aintpupython101

```
# Utility Functions
def evaluate_model(model, X_test, y_test,
model_name="Model"):
"""
Fits a given model to the test data, computes predictions,
and returns accuracy score and confusion matrix.
"""
y_pred = model.predict(X_test)
acc = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)
print(f"{model_name} Accuracy: {acc:.4f}")
print(classification_report(y_test, y_pred))
return acc, cm
```

https://tinyurl.com/aintpupython101

```
def plot_confusion_matrix(cm, labels=['No Risk', 'Risk'],
title="Confusion Matrix"):
"""
Plots a confusion matrix with the given labels.
"""
disp = sns.heatmap(cm, annot=True, fmt='g', cmap='Blues',
xticklabels=labels, yticklabels=labels)
plt.title(title)
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

https://tinyurl.com/aintpupython101

```
def plot_bar_comparison(df, x_col, y_col, title, x_label,
y_label):
"""
Plots a horizontal bar chart for comparison of models based
on a specified metric.
"""
fig = px.bar(df, y=y_col, x=x_col, color=y_col, text=x_col,
orientation='h', title=title)
fig.update_layout(xaxis_title=x_label, yaxis_title=y_label)
fig.show()
```

https://tinyurl.com/aintpupython101

```
# Modeling
models = {
   "Decision Tree": DecisionTreeClassifier(random_state=1),
   "Random Forest":
     RandomForestClassifier(n_estimators=100,
     random_state=1),
   "Logistic Regression":
     LogisticRegression(random_state=1),
   "XGBoost Classifier": XGBClassifier(random_state=1,
     use_label_encoder=False, eval_metric='mlogloss'),
   "Support Vector Machine": svm.SVC(kernel='linear',
     random_state=1),
   "Neural Network": MLPClassifier(random_state=1,
     max_iter=200)
   }
```

https://tinyurl.com/aintpupython101

```
results = []
confusion_matrices = {}

# Train, evaluate, and store results
for name, model in models.items():
model.fit(X_train, y_train)
acc, cm = evaluate_model(model, X_test, y_test,
model_name=name)
results.append((name, acc))
confusion_matrices[name] = cm
```

https://tinyurl.com/aintpupython101

```
# Compare Model Performance
results_df = pd.DataFrame(results, columns=['Model',
'Accuracy'])
results_df['Accuracy'] = results_df['Accuracy'] * 100
results_df = results_df.sort_values(by='Accuracy',
ascending=False)

print("\nModel Performance Comparison:")
print(results_df)
```

```
results_df['Text'] = results_df.apply(lambda row:
f"{row['Model']}: {row['Accuracy']:.2f}%", axis=1)
plot_bar_comparison(results_df, x_col="Accuracy",
y_col="Model",
title="Model Accuracy Comparison",
x_label="Accuracy (%)", y_label="Model")
```

https://tinyurl.com/aintpupython101

```
# Error Analysis (FP, FN, TN)
analysis_data = []
for name, cm in confusion_matrices.items():
TN, FP = cm[0,0], cm[0,1]
FN, TP = cm[1,0], cm[1,1]
analysis_data.append([name, FP, FN, TP])

analysis_df = pd.DataFrame(analysis_data, columns=['Model',
'False Positives', 'False Negative', 'True Negative'])
```

https://tinyurl.com/aintpupython101

```
# Visualize False Positives
fp_df = analysis_df[['Model', 'False
Positives']].sort_values(by='False Positives',
ascending=False)
fp_df['Text'] = fp_df.apply(lambda row: f"{row['Model']}:
{row['False Positives']}", axis=1)
plot_bar_comparison(fp_df, x_col='False Positives',
y_col='Model',
title="False Positives Comparison",
x_label="False Positives", y_label="Model")
```

https://tinyurl.com/aintpupython101

```
# Visualize False Negatives
fn_df = analysis_df[['Model', 'False
Negative']].sort_values(by='False Negative',
ascending=False)
fn_df['Text'] = fn_df.apply(lambda row: f"{row['Model']}:
{row['False Negative']}", axis=1)
plot_bar_comparison(fn_df, x_col='False Negative',
y_col='Model',
title="False Negatives Comparison",
x_label="False Negative", y_label="Model")
```

https://tinyurl.com/aintpupython101

```
# Visualize True Negatives
tn_df = analysis_df[['Model', 'True
Negative']].sort_values(by='True Negative',
ascending=False)
tn_df['Text'] = tn_df.apply(lambda row: f"{row['Model']}:
{row['True Negative']}", axis=1)
plot_bar_comparison(tn_df, x_col='True Negative',
y_col='Model',
title="True Negative (Correct Risk Predictions)
Comparison",
x_label="True Negative", y_label="Model")
```

https://tinyurl.com/aintpupython101

```
# Feature Importance (From Random Forest)
if 'Random Forest' in models:
   rfc = models['Random Forest']
   importances = rfc.feature_importances_
   indices = np.argsort(importances)[::-1]
   names = [X.columns[i] for i in indices]

     fig = px.bar(x=names, y=importances[indices],
     title="Feature Importance (Random Forest)")
   fig.update_xaxes(tickangle=90)
   fig.show()
```

https://tinyurl.com/aintpupython101

NVIDIA Developer Program

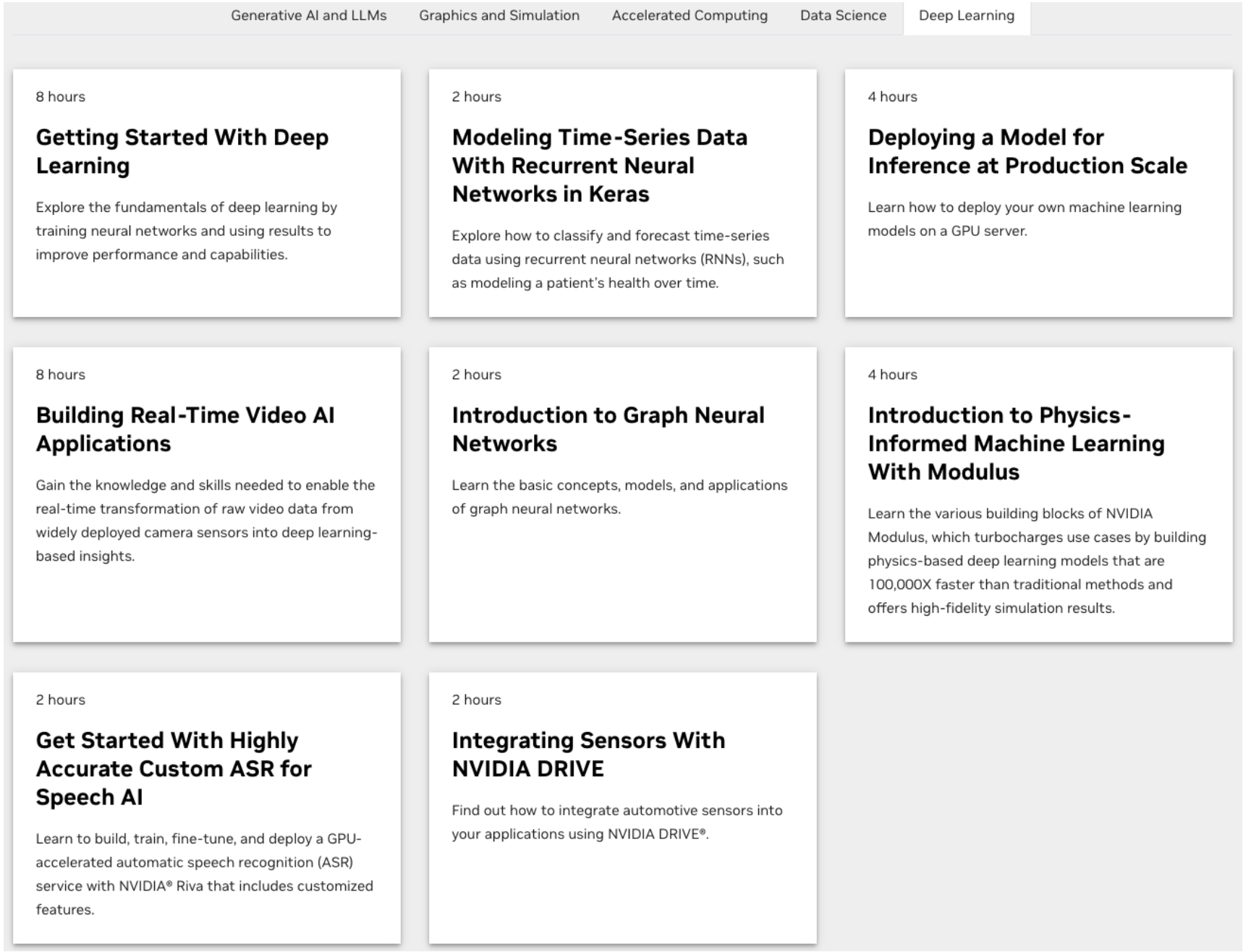
https://developer.nvidia.com/join-nvidia-developer-program

NVIDIA Deep Learning Institute (DLI)

https://learn.nvidia.com/

Join the NVIDIA Developer Program

take one of the complimentary technical self-paced courses (worth up to $90)



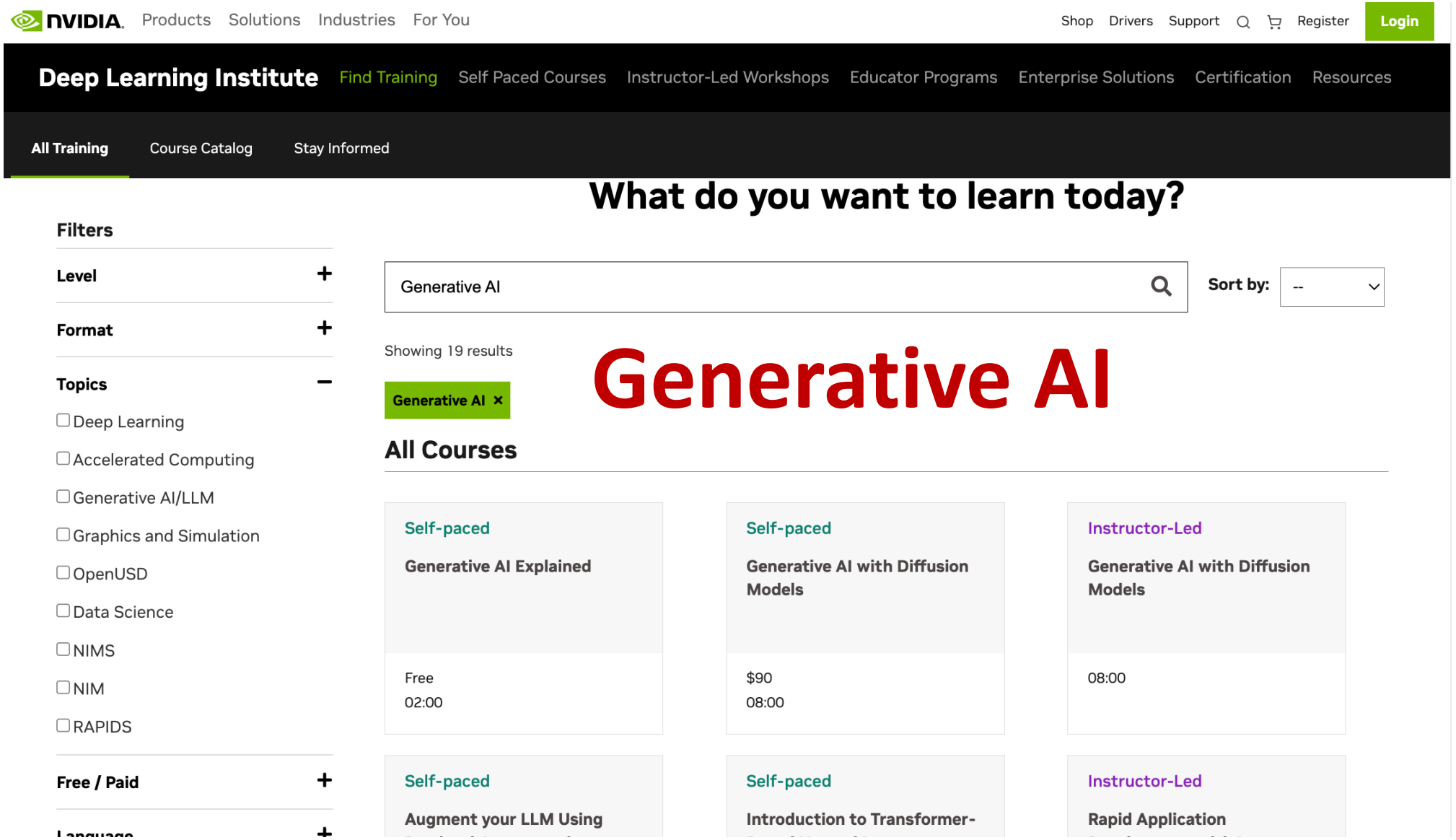
https://developer.nvidia.com/join-nvidia-developer-program

NVIDIA Deep Learning Institute (DLI)

Self-Paced Course **Generative AI Explained** Free 2 hours	Self-Paced Course **Getting Started With Deep Learning** Certificate available $90 8 hours	Instructor-Led Workshop **Fundamentals of Deep Learning** Certificate available $500 8 hours	Self-Paced Course **Introduction to Transformer-Based Natural Language Processing** Certificate available $30 6 hours
Self-Paced Course **Building RAG Agents With LLMs** Certificate available Free 8 hours	Instructor-Led Workshop **Building RAG Agents With LLMs** Certificate available $500 8 hours	Self-Paced Course **Generative AI with Diffusion Models** Certificate available $90 8 hours	Instructor-Led Workshop **Generative AI with Diffusion Models** Certificate available $500 8 hours

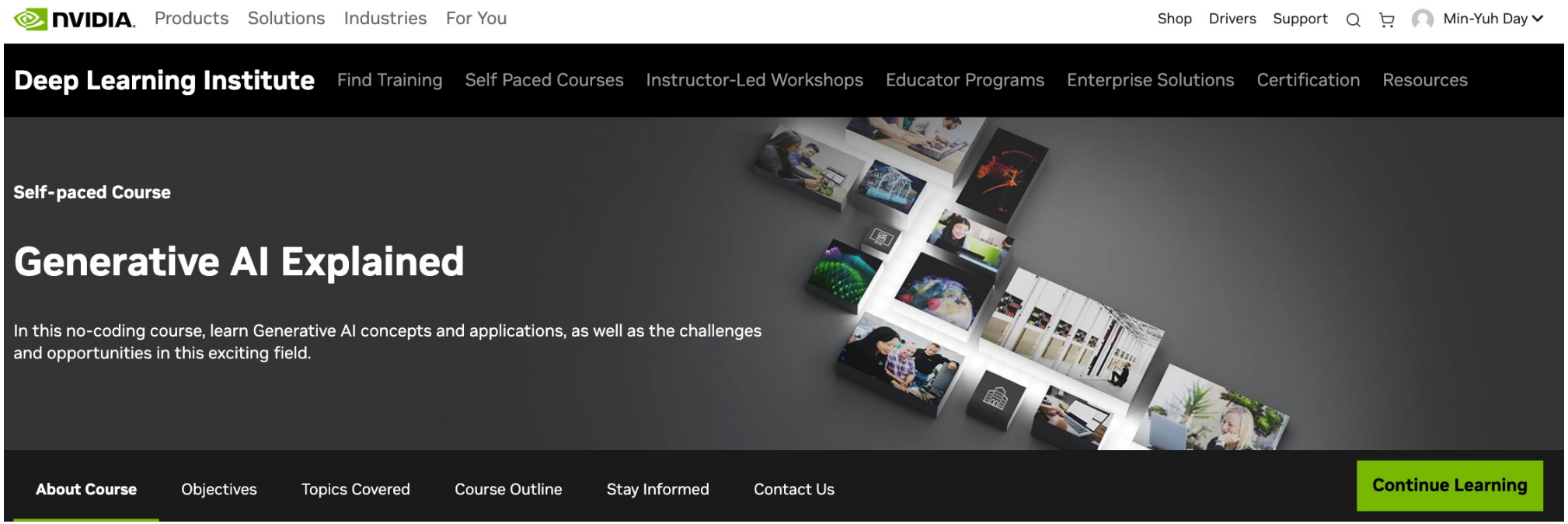
https://www.nvidia.com/en-us/learn/learning-path/generative-ai-llm/

NVIDIA Deep Learning Institute (DLI)



https://learn.nvidia.com/en-us/training/find-training?q=Generative+AI

Generative AI Explained



About this Course

Generative AI describes technologies that are used to generate new content based on a variety of inputs. In recent time, Generative AI involves the use of neural networks to identify patterns and structures within existing data to generate new content. In this course, you will learn Generative AI concepts, applications, as well as the challenges and opportunities in this exciting field.

Learning Objectives

Upon completion, you will have a basic understanding of Generative AI and be able to more effectively use the various tools built on this

Course Details

Duration: 02:00

Price: Free

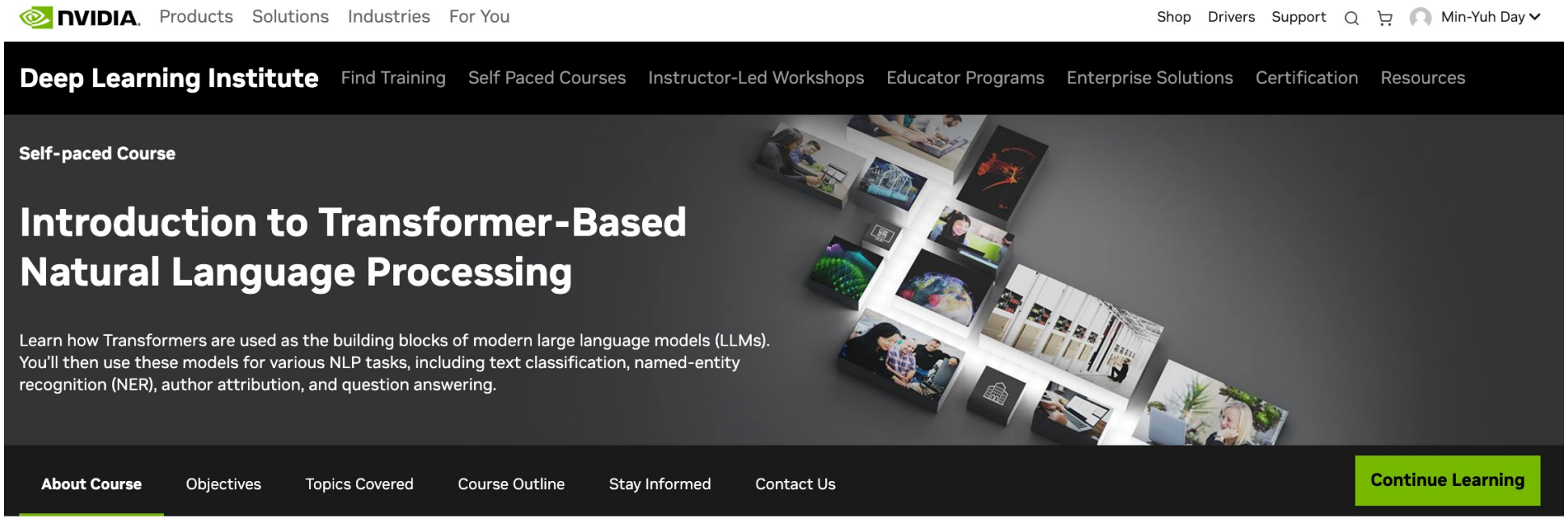
Level: Technical - Beginner

Subject: Generative AI/LLM

Language: English

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-15+V1

Introduction to Transformer-Based Natural Language Processing



About this Course

Large Language Models (LLMs), or Transformers, have revolutionized the field of natural language processing (NLP). Driven by recent advancements, applications of NLP and generative AI have exploded in the past decade. With the proliferation of applications like chatbots and intelligent virtual assistants, organizations are infusing their businesses with more interactive human-machine experiences. Understanding how Transformer-based large language models (LLMs) can be used to manipulate, analyze, and generate text-based data is essential. Modern pre-trained LLMs can encapsulate the nuance, context, and sophistication of language, just as humans do. When fine-tuned and deployed correctly, developers can use these LLMs to build powerful NLP applications that provide natural and seamless human-computer interactions within chatbots, AI voice agents, and more. In this course, you'll learn how Transformers are used as the building blocks of modern large language models (LLMs). You'll then use these models for various NLP

Course Details

Duration: 06:00

Price: $30

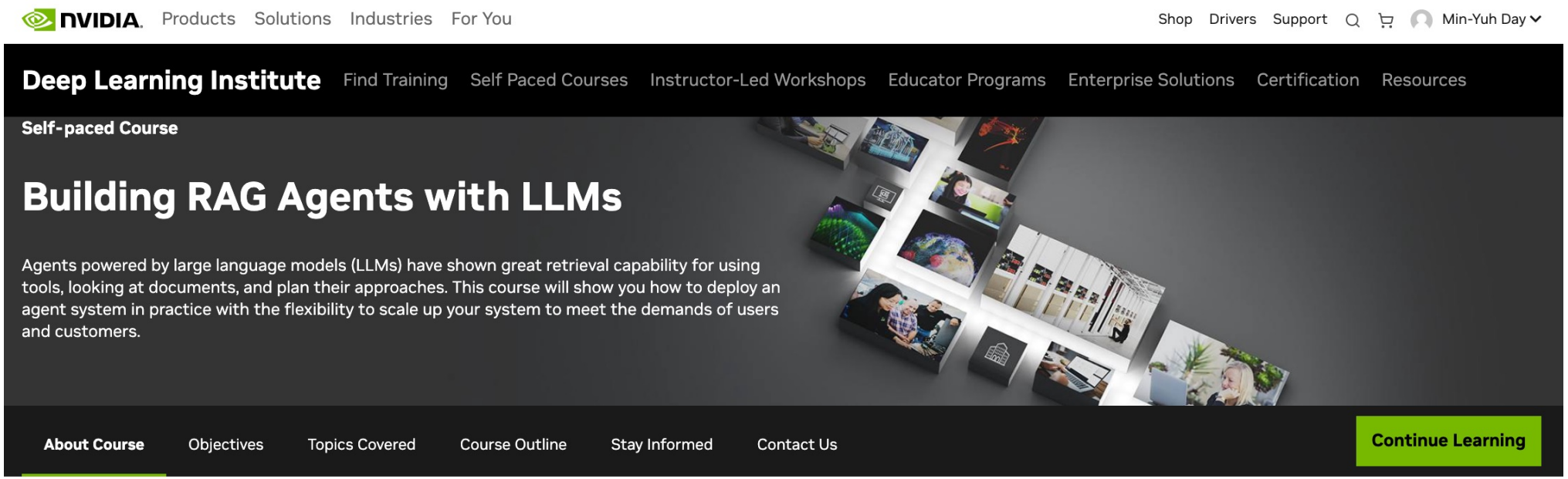
Level: Technical - Beginner

Subject: Generative AI/LLM

Language: English

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-08+V1

Building RAG Agents with LLMs



About this Course

This course is free for a limited time.

The evolution and adoption of large language models (LLMs) have been nothing short of revolutionary, with retrieval-based systems at the forefront of this technological leap. These models are not just tools for automation; they are partners in enhancing productivity, capable of holding informed conversations by interacting with a vast array of tools and documents. This course is designed for those eager to explore the potential of these systems, focusing on practical deployment and the efficient implementation required to manage the considerable demands of both users and deep learning models. As we delve into the intricacies of LLMs, participants will gain insights into advanced orchestration techniques that include internal reasoning, dialog management, and effective tooling strategies.

Course Details

Duration: 08:00

Price: Free

Level: Technical - Intermediate

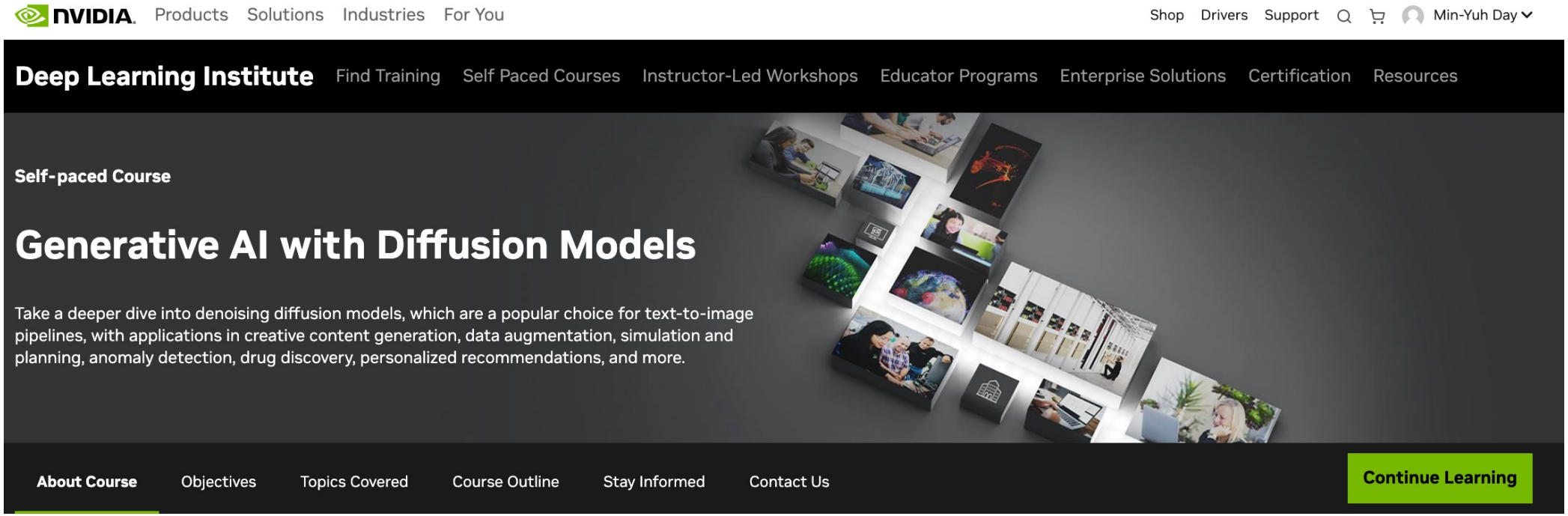
Subject: Generative AI/LLM

Language: English

Course Prerequisites:
Introductory deep learning knowledge, with comfort

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-15+V1

Generative AI with Diffusion Models



About this Course

Thanks to improvements in computing power and scientific theory, generative AI is more accessible than ever before. Generative AI plays a significant role across industries due to its numerous applications, such as creative content generation, data augmentation, simulation and planning, anomaly detection, drug discovery, personalized recommendations, and more. In this course, learners will take a deeper dive into denoising diffusion models, which are a popular choice for text-to-image pipelines.

Course Details

Duration: 08:00

Price: $90

Subject: Generative AI/LLM

Language: English

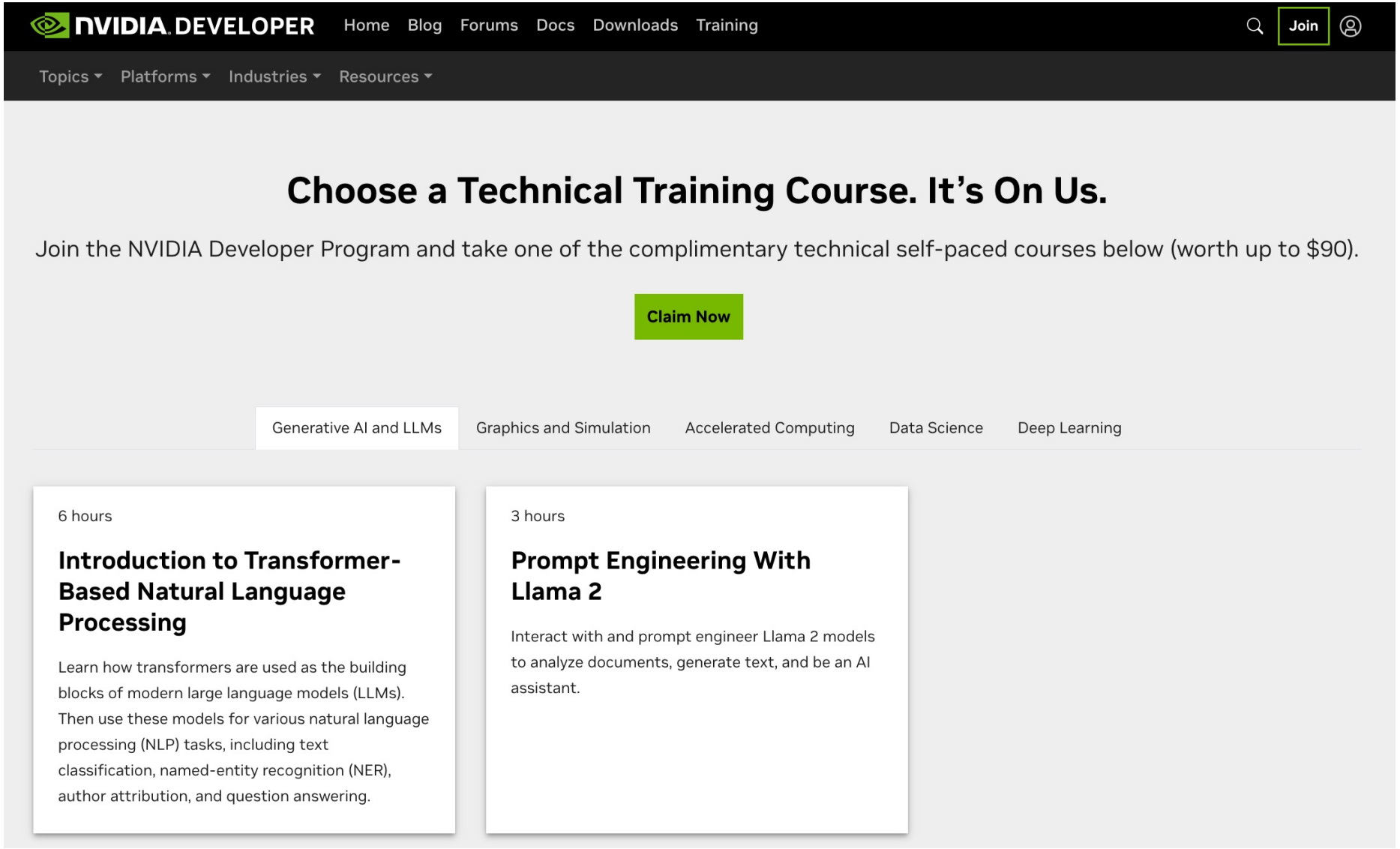
Course Prerequisites:
A basic understanding of Deep Learning Concepts.

Learning Objectives

https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-14+V1

Join the NVIDIA Developer Program

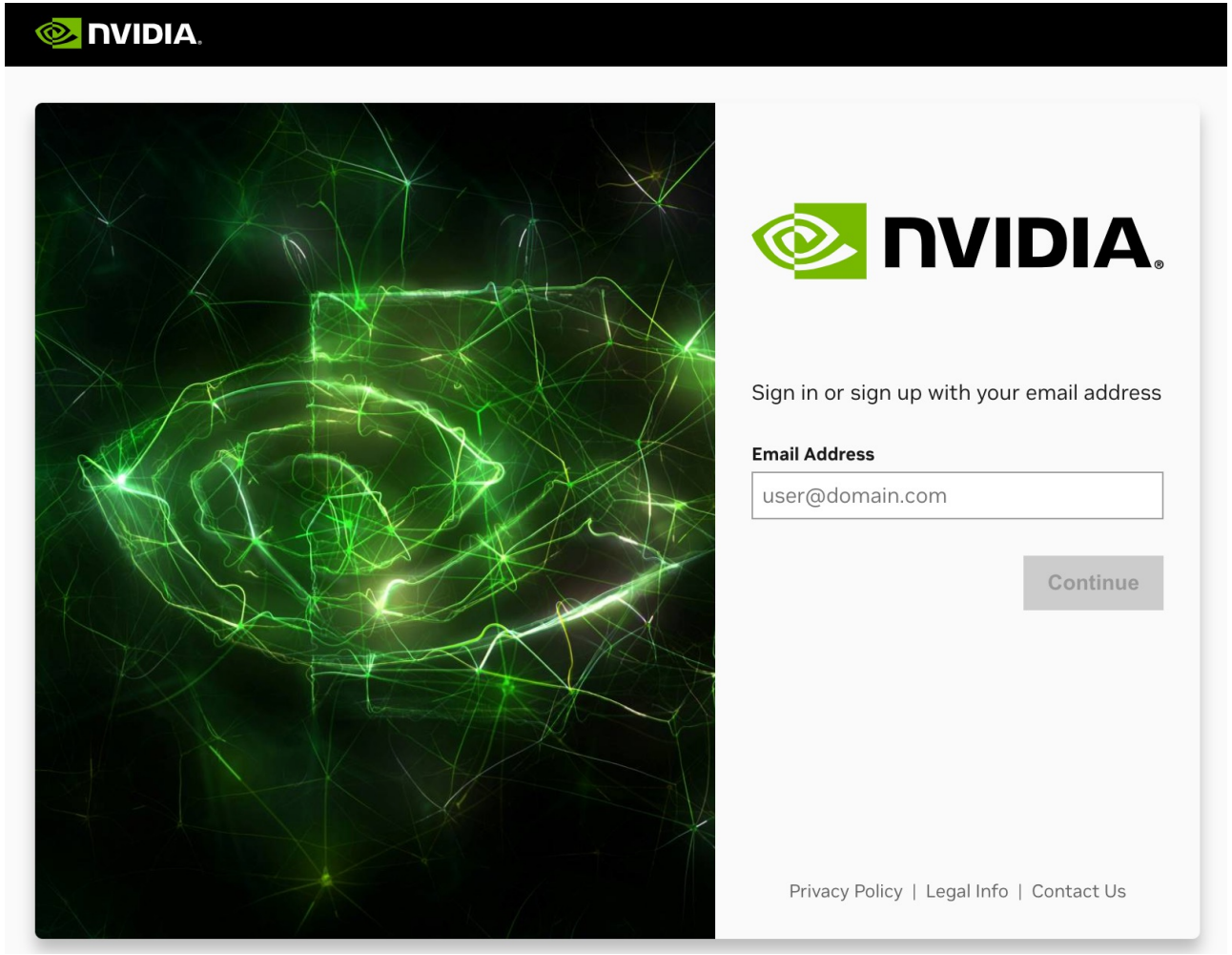
take one of the complimentary technical self-paced courses (worth up to $90)



https://developer.nvidia.com/join-nvidia-developer-program

Join the NVIDIA Developer Program

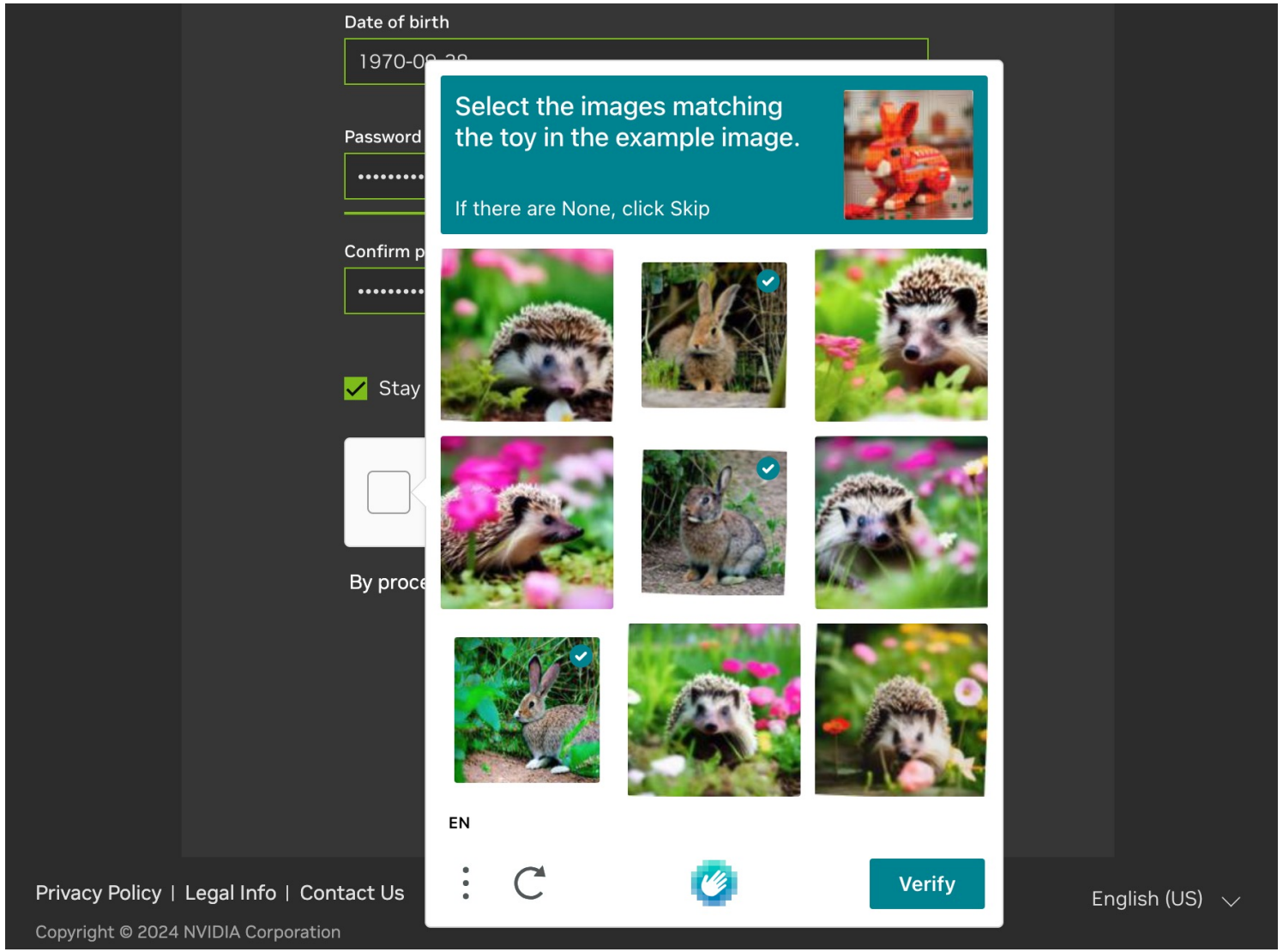
take one of the complimentary technical self-paced courses (worth up to $90)



https://developer.nvidia.com/join-nvidia-developer-program

Join the NVIDIA Developer Program

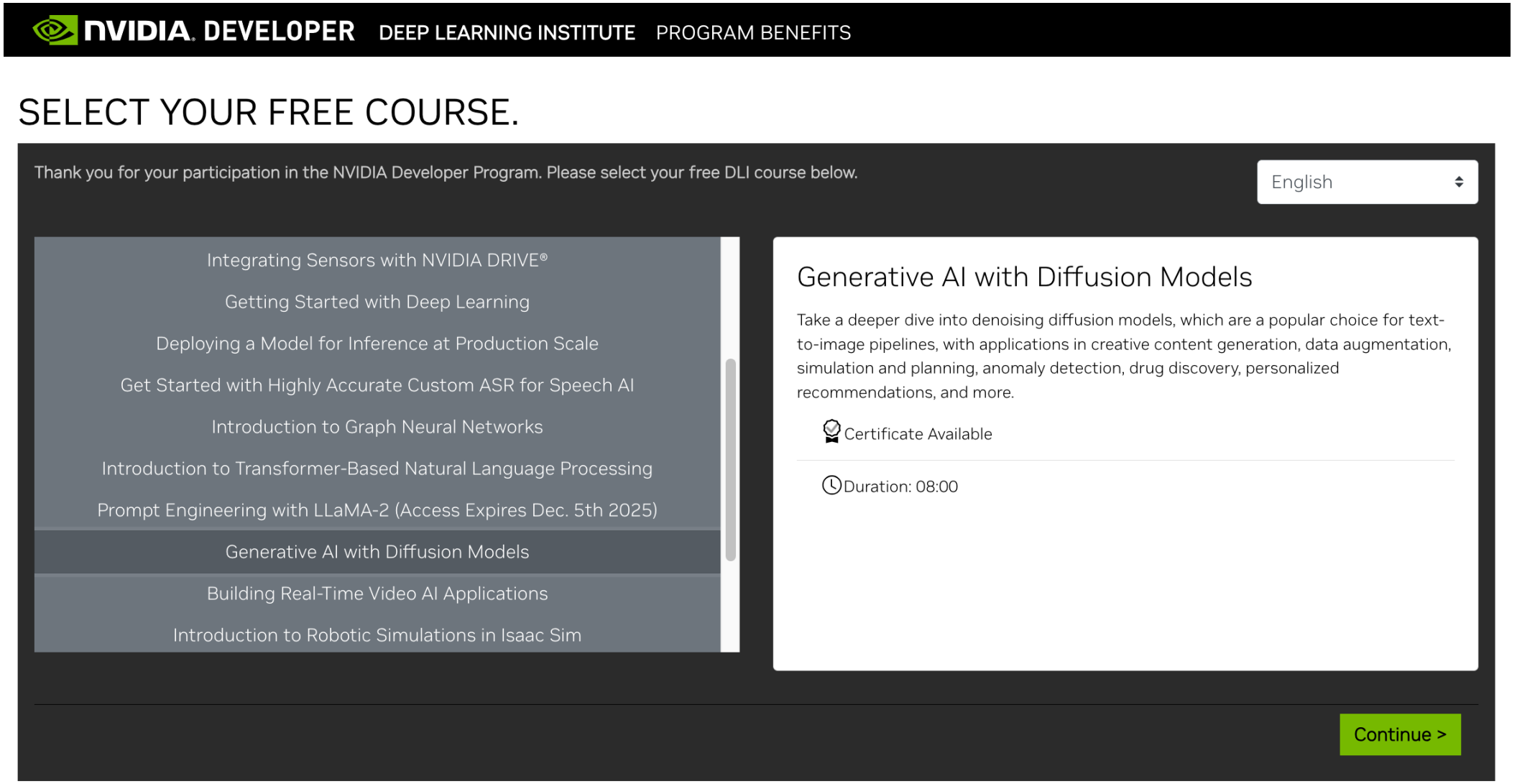
take one of the complimentary technical self-paced courses (worth up to $90)



https://developer.nvidia.com/join-nvidia-developer-program

Join the NVIDIA Developer Program

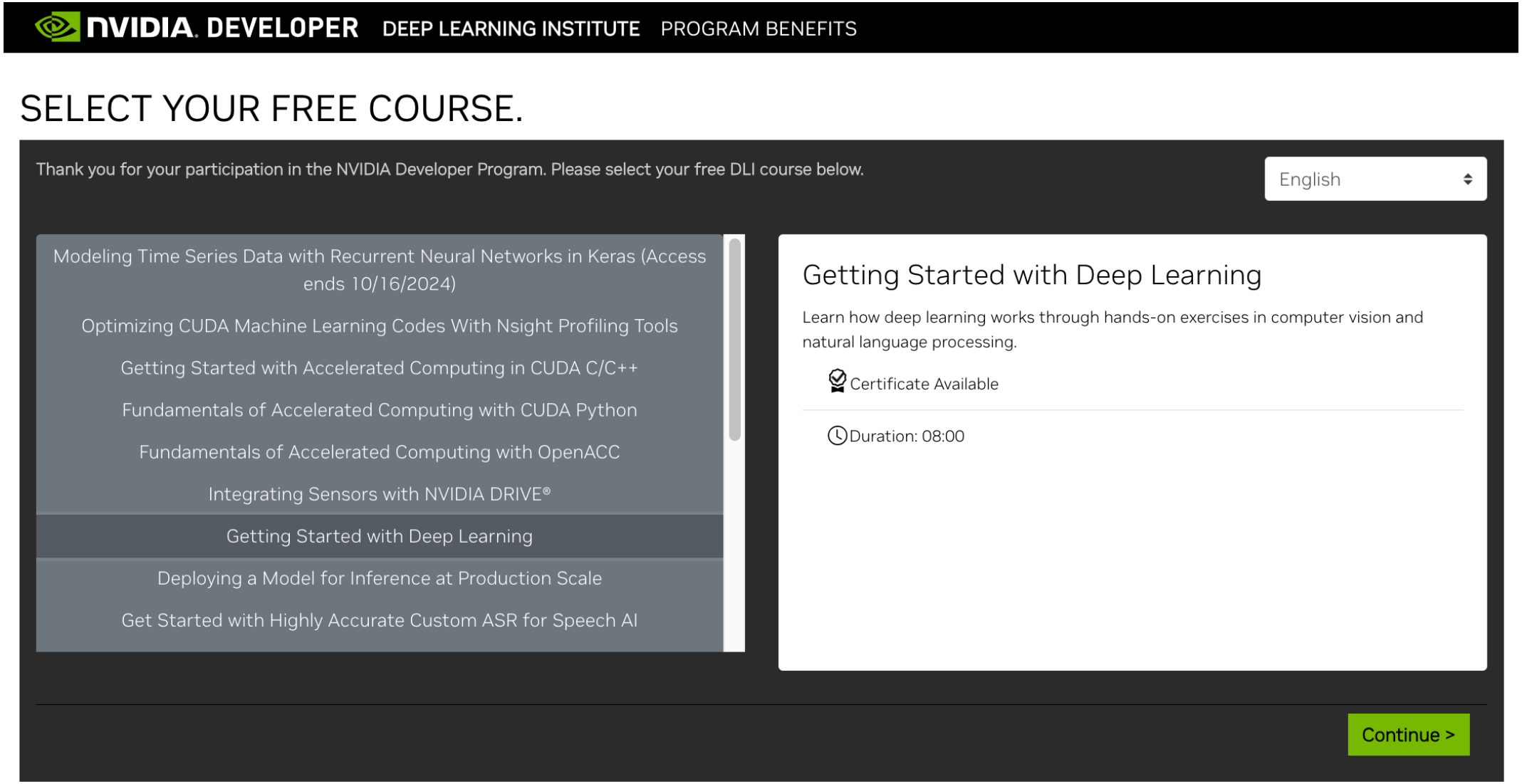
take one of the complimentary technical self-paced courses (worth up to $90)



https://developer.nvidia.com/join-nvidia-developer-program

Join the NVIDIA Developer Program

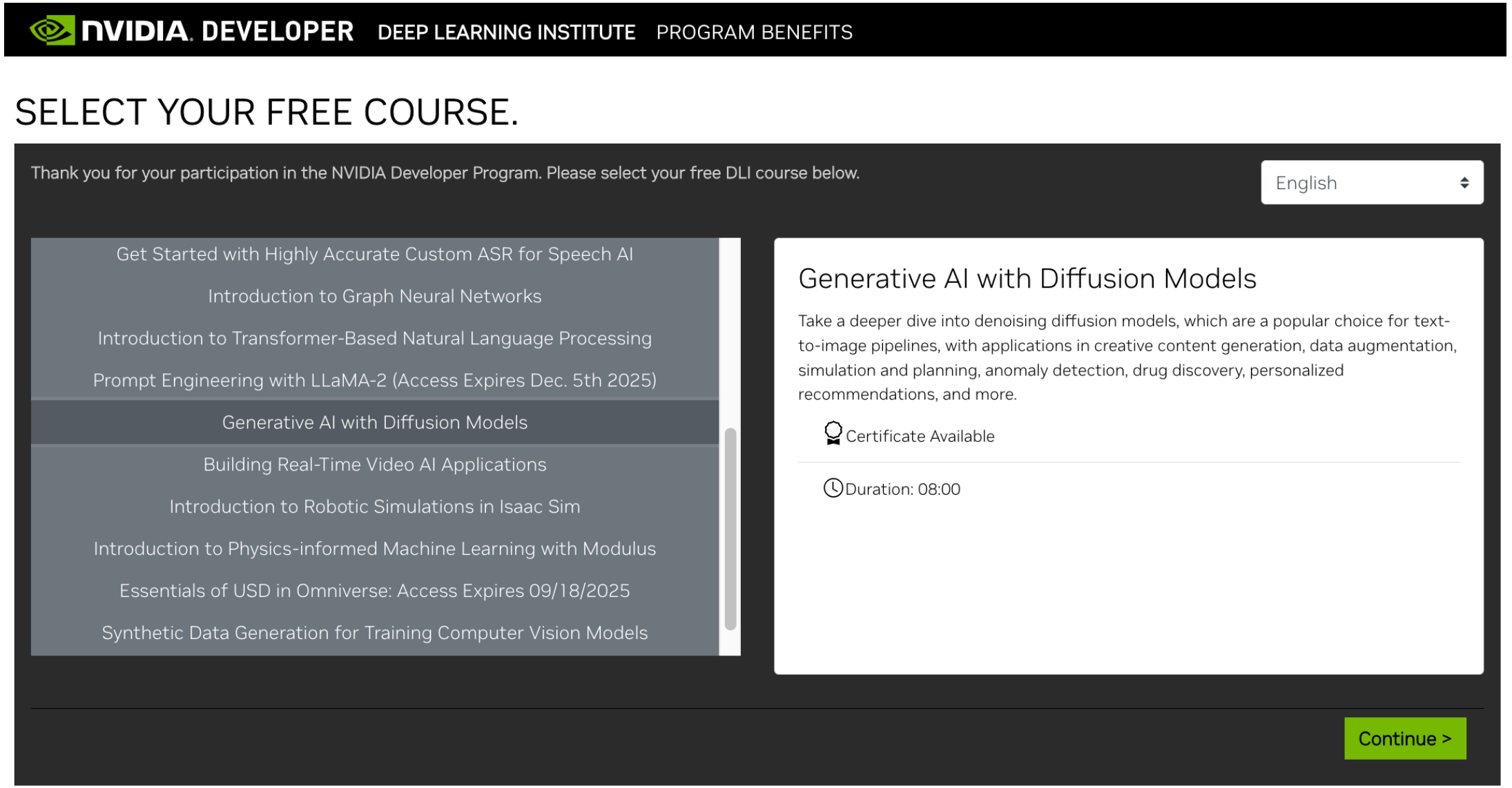
take one of the complimentary technical self-paced courses (worth up to $90)



https://developer.nvidia.com/join-nvidia-developer-program

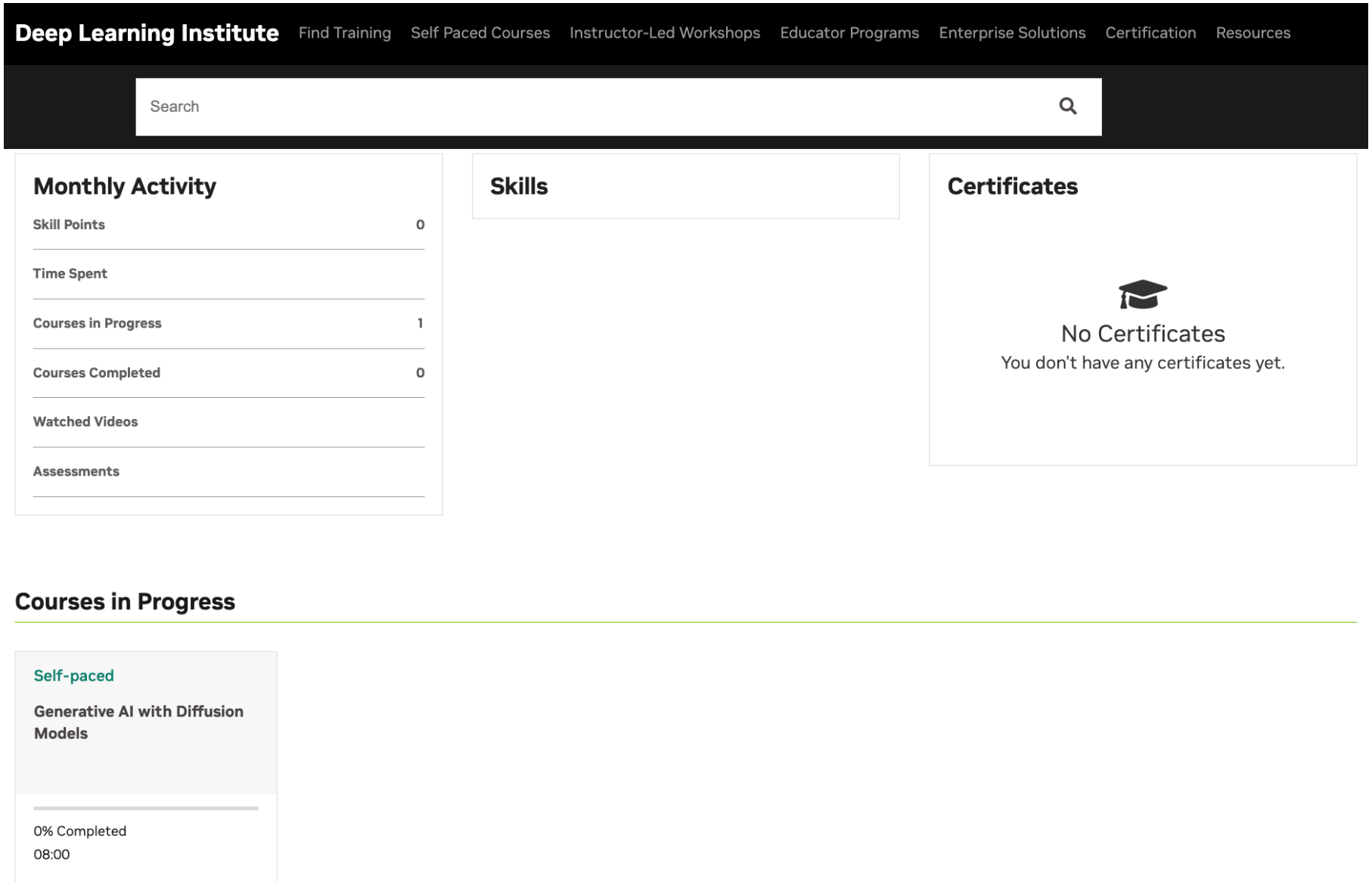
Join the NVIDIA Developer Program

take one of the complimentary technical self-paced courses (worth up to $90)



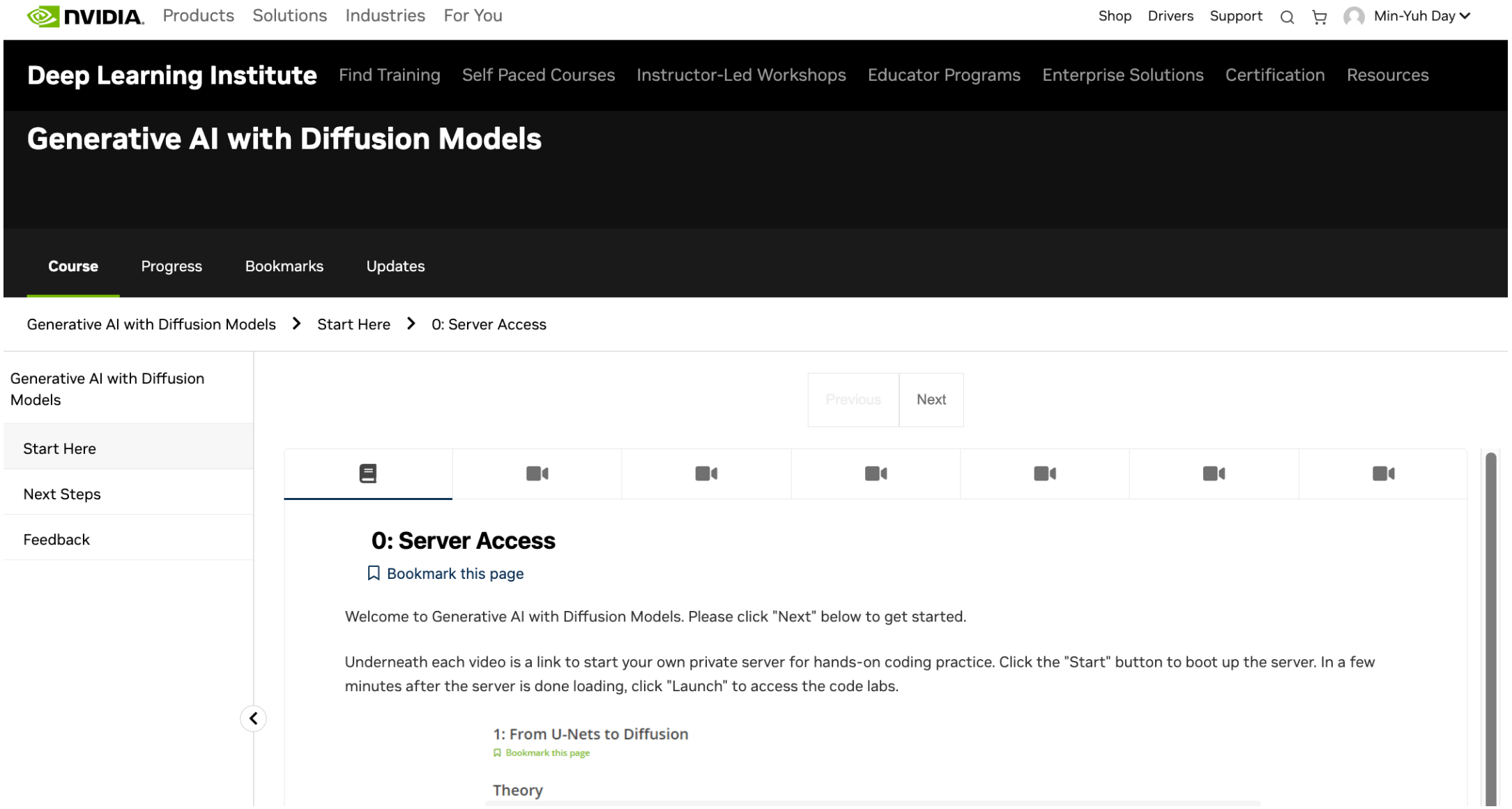
https://developer.nvidia.com/join-nvidia-developer-program

NVIDIA Deep Learning Institute (DLI)



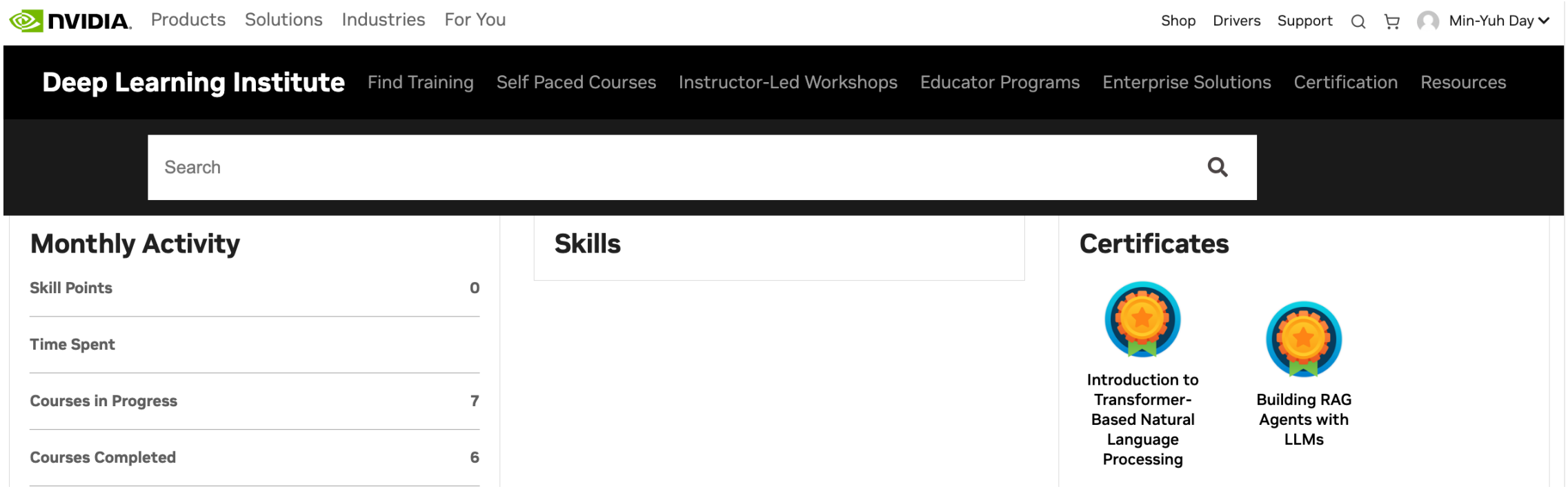
https://learn.nvidia.com/my-learning

NVIDIA Deep Learning Institute (DLI)



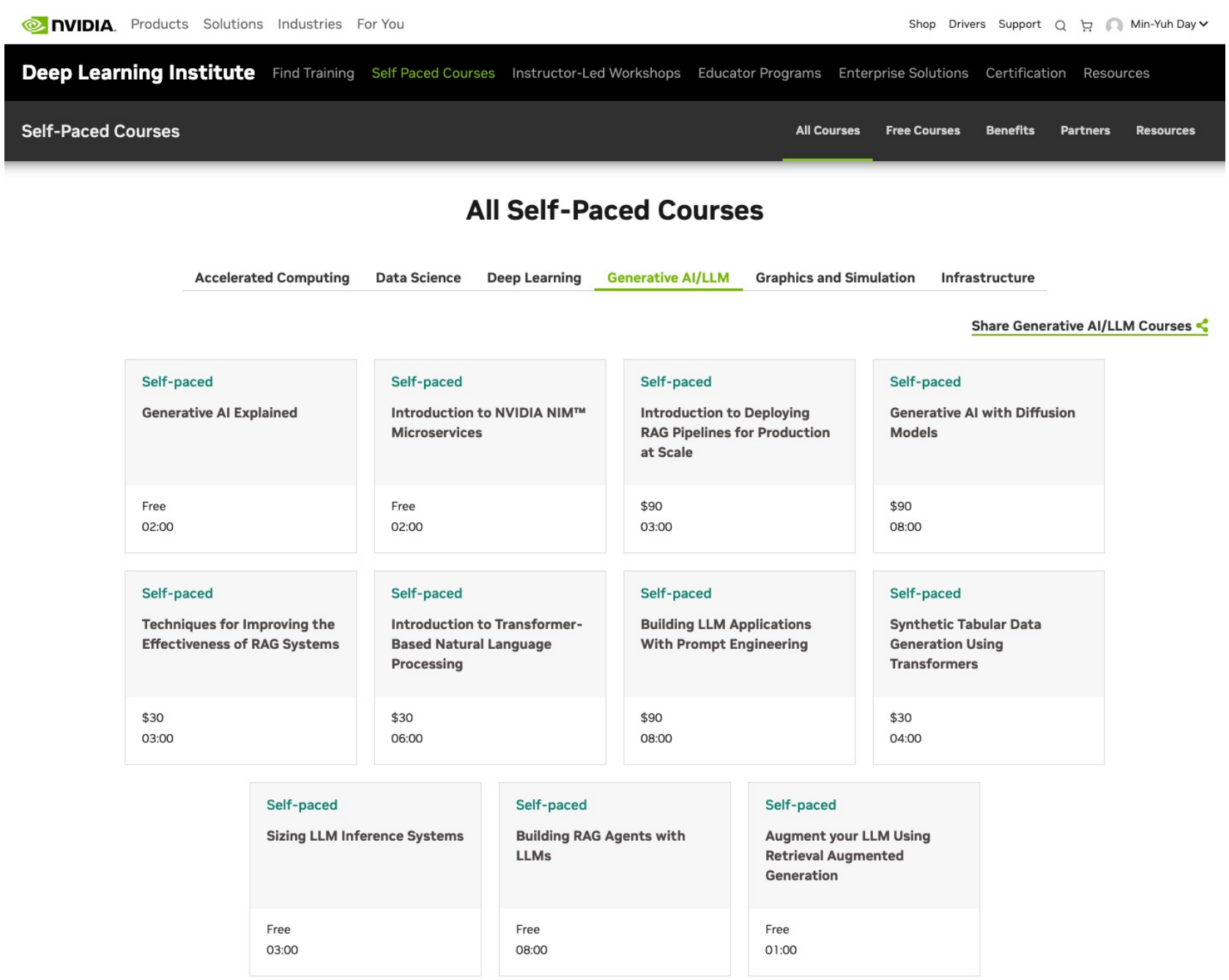
https://learn.nvidia.com/my-learning

Deep Learning Institute (DLI)



https://learn.nvidia.com/my-learning

NVIDIA Deep Learning Institute (DLI)



https://learn.nvidia.com/en-us/training/self-paced-courses

NVIDIA

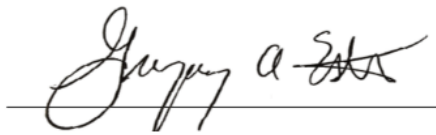
Certificate of Competency

This certificate is awarded to

Min-Yuh Day

for demonstrating competence in the completion of

Introduction to Transformer-Based Natural Language Processing



Greg Estes

Vice President, NVIDIA

Issue Date: : December 5, 2024

Certification ID: twpWsrB4TDCQ0ErgAoEt6w |

https://learn.nvidia.com/certificates?id=twpWsrB4TDCQ0ErgAoEt6w/courses/course?course_id=course-v1:DLI+S-FX-08+V1

https://learn.nvidia.com/certificates?id=twpWsrB4TDCQ0ErgAoEt6w

NVIDIA

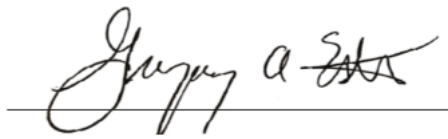
Certificate of Completion

This certificate is awarded to

Min-Yuh Day

for successfully completing

Building RAG Agents with LLMs



Greg Estes

Vice President, NVIDIA

Issue Date: : December 8, 2024

Certification ID: ed-qOCIMQaatzU8SNUNxgw |

https://learn.nvidia.com/certificates?id=ed-qOCIMQaatzU8SNUNxgw/courses/course?course_id=course-v1:DLI+S-FX-15+V1

https://learn.nvidia.com/certificates?id=ed-qOCIMQaatzU8SNUNxgw