

הגנת סייבר בשילוב Gemini API

כתב: שלום ואנונו, מורה להנדסת תוכנה חלופת סייבר במקיף ח' ראשון לציון

השתלמות ייחודית בהנחיית [מר שלום ואנונו](#) עסקה בשילוב מודלי AI בסביבות פיתוח מקומיות: "Local Ollama & Gemini API". ההשתלמות נפתחה ב-22 ביוני, 2025, בשנה"ל תשפ"ה, במתכונת היברידית (זום בשילוב מפגש סיכום פיזי במרכז הארץ) וארכה 30 שעות אקדמיות (15 שעות עיוניות ו-15 שעות אסינכרוניות) עם ציון מוכר למורי "עוז לתמורה" ו"אופק חדש".

קהל היעד כלל מורי תיכון המלמדים במסלול הנדסת תוכנה וסייבר, עם ותק של לפחות שנתיים בהוראת הגשות פרויקטים. חשיבות ההשתלמות היא הקניית ידע תאורטי, ולא פחות חשוב, ידע מעשי ופרקטי Hand-On ביישום והוראה לתלמידים, בליווי מצגות עשירות ונוחות להוראת הנושא.

מטרות ההשתלמות

מטרת-העל של ההשתלמות היא שילוב בינה מלאכותית בפרויקטים במסגרת התמחות [הגנת סייבר ומערכות הפעלה](#) במגמת הנדסת תוכנה וסייבר, ההשתלמות נועדה להעניק למורים את הידע והכלים המעשיים הדרושים לשילוב מוצלח של בינה מלאכותית (AI) בפרויקטים חינוכיים החל משנת הלימודים תשפ"ו. הדגש הוא על הקניית יכולות יישומיות שיאפשרו למורים ולתלמידיהם:

- הבנה יסודית של מושגי ה-AI: הכרת עקרונות הבינה המלאכותית, סוגיה השונים והפוטנציאל הטמון בה, תוך פישוט והנגשה התכנים.
- שליטה בכלי AI בסיסיים ומתקדמים: למידת דרך העבודה עם ממשקי AI נפוצים (כמו Google API), כתיבת פרומפטים יעילים, ואף הקמת סביבות AI מקומיות (Local LLMs) לצורך ניסוי ופיתוח.
- פיתוח חשיבה יצירתית ופתרון בעיות באמצעות AI: הבנה כיצד ניתן לרתום את הבינה המלאכותית לכתיבת קוד, יצירת תכנים וביצוע משימות מורכבות, לפתיחת אופקים חדשים לפרויקטים לימודיים.
- הבנה ויישום של סוכני AI: הכרת עולם "סוכני ה-AI", ולמידת העקרונות לבנייתם על מנת שיוכלו לשמש ככלים לביצוע משימות אוטומטיות ומורכבות בפרויקטים.
- חיבור AI למאגרי מידע חיצוניים: לימוד כיצד להקנות לבינה המלאכותית את היכולת לאחזר מידע ממקורות חיצוניים (כמו מסמכי PDF) באמצעות טכניקות כמו RAG, יכולת שתאפשר פיתוח פרויקטים מבוססי ידע עדכני ומדויק.
- הנגשת AI לתלמידים: הקנייה למורים את הביטחון והיכולת להדריך תלמידים בשימוש בבינה מלאכותית בפרויקטים שלהם, להכנים לאתגרי העולם הטכנולוגי העתידי, ולטפח אוריינות דיגיטלית מתקדמת.
- פרטי ההשתלמות מוצגים להלן. כפי שניתן לראות, בסיום ההשתלמות, המורים היו מצוידים בידע פרקטי ובכלים יישומיים, שיאפשרו להם להוביל פרויקטים חדשניים ומרתקים המשלבים בינה מלאכותית, וכך להפוך את תהליך הלמידה לרלוונטי, עדכני ומועיל יותר עבור התלמידים.

מפגש ראשון: היכרות עם עולם ה-AI

מפגש מקוון ראשון: היכרות עם עולם ה-AI

• שיחת פתיחה והנחיות:

פתחנו את ההשתלמות בסקירה מקיפה של התכנים, המטרות והמתודולוגיה. הסברנו כיצד ניתן למקסם את הלמידה ולרתום את הכלים שהוצגו לטובת פרויקטים אישיים ומקצועיים.

• מבוא לבינה מלאכותית:

הבנו מהי בינה מלאכותית לעומק, היכרות עם הטכנולוגיה. סקרנו את העקרונות הבסיסיים, סוגי ה-AI השונים ואת האופן שבו טכנולוגיה זו שינתה את חיינו ואת עולם התעשייה.

• פתיחת משתמש Google API וביצוע Prompt עם Python:

למדנו כיצד להגדיר סביבת עבודה עם ממשק התכנות של גוגל (Google API) עבור AI. הדגמנו כיצד לבצע "פרומפטים" (הוראות/שאלות למודל AI) ראשוניים באמצעות קוד פשוט בשפת Python, כדי להתחיל לתקשר עם המודלים.



Made with Canva
הנחיות ראשונות Google API

המפגש הראשון של ההשתלמות נפתח בשיחת פתיחה חווייתית ובהנחיות ממוקדות, אשר הזמינו את המשתתפים להצטרף למסע רווי חדשנות וחקירה של תחום הבינה המלאכותית, תוך הדגשה כיצד יוכלו למצות באופן מרבי את תכני ההשתלמות. לאחר מכן, הוצג מבוא תמציתי לבינה מלאכותית, בו הוצגו מושגי יסוד והוסבר באופן פשוט הרציונל מאחורי טכנולוגיות AI, לרבות התייחסות למושגים כמו למידת מכונה, למידה עמוקה ומודלים שפתיים גדולים (LLMs).

מטרות הלמידה:

- להבין את מושגי היסוד בבינה מלאכותית: סוגי LLMs, AI ומודלים בענן.
- להכיר את ממשק Google API ולפתוח חשבון פרויקט למטרות ניסוי.
- להקים סביבת Python מקומית ולבצע קריאת API ראשונה ("Hello AI").
- לפתח מודעות לפוטנציאל הפדגוגי של שילוב AI בפרויקטי תלמידים.

בחלק המעשי של השיעור למדו המורים כיצד להקים משתמש ב-Google API להפעלת מודלים גנרטיביים, וזכו להדגמה חיה של ניסוח Prompts באמצעות שפת Python, הכוללת טעינת מפתח API והרצת פקודות ראשוניות ליצירת תוכן באמצעות מודל Gemini.

המפגש השני התמקד בשיפור יכולת האינטראקציה עם מודלים שפתיים: בראש ובראשונה הוצגו עקרונות הכתיבה של פרומפט איכותי: חמישה שלבים שכללו הגדרת התפקיד (Role), המשימה (Task), ההקשר (Context), המגבלות (Constraints) והפורמט (Format), במטרה לחדד את הבקשה ולמנוע אי-הבנות מצד ה-AI.

Gemini

<https://gemini.google.com/>

<https://aistudio.google.com/>

מפגש ראשון: היכרות עם עולם ה-AI

קבלת Gemini API חינם

<https://aistudio.google.com/app/apikey>

Project number	Project name	API key	Created	Plus
...3021	Gemini API	...PILKA	Mar 4, 2025	Flag Set up billing View usage data

מפגש שני: שליטה על ה-AI באמצעות פרומפטים

- עבודה עם Google CLI (Command Line Interface):

הכרנו את ה-CLI של גוגל – כלי שורת פקודה שאפשר אינטראקציה ישירה עם שירותי ה-AI של גוגל מהמחשב המקומי, ללא צורך בסביבת פיתוח גרפית.
- עקרונות לכתיבת פרומפט איכותי:

כיצד לגרום למודל ה-AI להבין בדיוק מה רצינו? למדנו את המתודולוגיות והטיפים לכתיבת פרומפטים יעילים, מדויקים וברורים, שהבטיחו קבלת תגובות אופטימליות מהבינה המלאכותית.
- כתיבת פרומפטים מתקדמים בעבודה מול API:

התקדמנו מעבר לפרומפטים בסיסיים ולמדנו טכניקות לכתיבת פרומפטים מורכבים וחכמים יותר, תוך שימוש בממשקי API כדי למצות את מלוא היכולות של המודלים.
- JSON, Example Prompt, Structure, Continues Prompt:

מטרות הלמידה

- להפעיל Google CLI ולהריץ פקודות AI ישירות מחלון ה-Terminal.
- לכתוב פרומפטים ברורים, מדויקים ומבוססי הקשר (Prompt Engineering).
- ליישם Example Prompting לשיפור עקביות התשובות.
- לנהל שיחות ממושכות בעזרת Continues Prompt ושימור הקונטקסט.
- להבין מבנה תקין של payload JSON להעברת פרמטרים מתקדמים למודל.

המשך המפגש התמקד בכתיבת פרומפטים מתקדמים בעבודה מול API, כולל הכנסת פרמטרים דינמיים, הבחנה בין zero-shot, one-shot ו-few-shot prompting וניסוח בקשות חכמות יותר ב-Python המתקשרות ישירות ל-Gemini API. בסיום המפגש, הוצג פרומט JSON, לצד הצגת דוגמת prompt מלאה הכוללת מבנה JSON ברור, שדות מקוונים ו-Continues Prompt, המבטיחים אחידות פלט, שליפה פשוטה ועיבוד נתונים יעיל בסביבות הפיתוח השונות.

מפגש שני: שליטה על ה-AI באמצעות פרומפטים

Installation

Get started by installing Gemini CLI.

Prompt Structure

משחק בעזרת Gemini API

```

import PromptFunction as AI

PROMPT = """
You are a participant in a game where I need to guess a
TARGET_WORD you describe in a sentence or two.
You cannot mention the TARGET_WORD or any of the
additional forbidden words.

TARGET_WORD: 'router'
additional forbidden words: internet, wifi, device, traffic
description:
"""

print(AI.ask_gemini(PROMPT))

```

This box directs network data between different connected systems in a home or office. It acts as a central hub, sending information packets where they need to go.

מפגש שלישי: פיתוח עם AI ומודלים מקומיים

פיתוח אפליקציות Vibe Coding (פיתוח Full Stack בשיתוף פרומט)

חקרנו את הקונספט המהפכני של "Vibe Coding" – פיתוח אפליקציות מלאות (Full Stack) באמצעות אינטראקציה עם AI דרך פרומפטים בלבד, ללא צורך בכתיבת קוד רב באופן ידני.

- כלים לכתיבת קוד עם בינה מלאכותית ואתרים:
- סקרנו כלים ואתרי עזר ששילבו AI כדי לסייע בפיתוח קוד, לייצר קטעי קוד, לתקן שגיאות ולזרז תהליכי פיתוח.
- הקמת (Local LLM מודל שפה גדול מקומי) עם תשתית: Ollama, למדנו כיצד להקים ולהריץ מודלי שפה גדולים (LLMs) על המחשב האישי שלנו, באמצעות פלטפורמת Ollama. זה איפשר עבודה עם מודלים חזקים ללא תלות בחיבור לאינטרנט או בשירותי ענן חיצוניים.
- ממשק קוד לעבודה מול Local LLM בשפת Python: לאחר הקמת המודל המקומי, למדנו כיצד "לתקשר" איתו ולשלוח לו פקודות ומשימות באמצעות קוד Python פשוט ויעיל.
- היכרות עם מודלים (Open Source קוד פתוח) וייחודם:

סקרנו את עולם מודלי ה-AI בקוד פתוח – מודלים שזמינים לשימוש, שינוי והפצה חופשית. הבנו את היתרונות הייחודיים שלהם, כגון גמישות, שקיפות ויכולת התאמה אישית.

המפגש השלישי התמקד בהקמת סביבה מקומית לעיבוד שפה גדולה באמצעות Ollama, החל מהתקנת השרת המקומי ועד להפעלת מודל ראשוני בסביבת הפיתוח של המורים, כך שיוכלו להריץ LLM ללא תלות באינטרנט ובענן. בהמשך, למדו המשתתפים כיצד להתחבר ל-L-Local דרך ממשק הקוד בפייתון – בין

אם באמצעות HTTP API פשוט או בעזרת הספרייה ollama ולשלוח אליו שאילתות מקוד מותאם, כולל שילוב עם LangChain לשרשור פעולות מתקדמות.

מטרות הלמידה

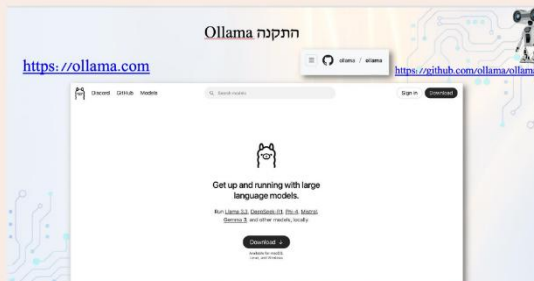
- להבין את יתרונות וחסרונות הפעלת LLM מקומי לעומת SaaS-AI בענן.
- להוריד, להתקין ולהריץ מודלים ב-Ollama (לדוגמה Llama 2 או Mistral-7B).
- לכתוב קוד Python המתחבר לתהליך המקומי דרך REST-API של Ollama.
- ליישם פיתוח אפליקציות Full-Stack בשיתוף Vibe Coding תוך שימוש ב-AI ל- front-end, back-end ושכבת DB-schema.
- להכיר כלים פתוחי-קוד כגון TabNine, GitHub Copilot, ו-AutoRegex המסייעים בתיקון, השלמה ובדיקת קוד.



Vibe Coding Wars

פיתוח קוד בעזרת בינה מלאכותית: Vibe Coding

מפגש שלישי: פיתוח עם AI ומודלים מקומיים



Ollama חתקנה

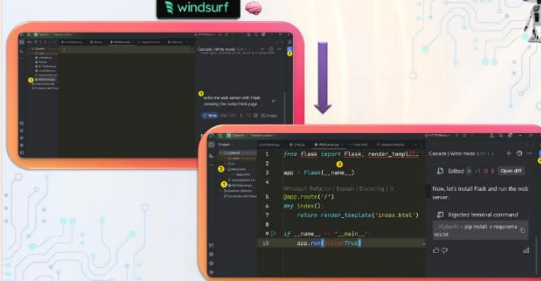
<https://ollama.com>

<https://github.com/ollama/ollama>

Get up and running with large language models.

For Llama 3.3, Gemma 3.1, and other models, locally.

Download



windsurf

```

1 from flask import Flask, render_template, request
2 app = Flask(__name__)
3
4 @app.route("/")
5 def index():
6     return render_template("index.html")
7
8 if __name__ == "__main__":
9     app.run(debug=True)
        
```

שימוש ישיר בספריית ollama

ספריית ollama מספקת דרך ישירה יותר לתקשר עם שרת Ollama.

התקנה:

```

1 import ollama
2
3 # Initialize the Ollama client
4 client = ollama.Client()
5
6 # Define the model and the input prompt
7 model = "gemma3" # Replace with your model name
8 prompt = "What is Python?"
9
10 # Send the query to the model
11 response = client.generate(model=model, prompt=prompt)
12
13 # Print the response from the model
14 print("Response from Ollama:")
15 print(response.response)
        
```

pip install ollama

בנוסף הוצגה סקירה של מודלים לשפה טבעית בעלי משקלים פתוחים (כגון LLaMA, Gemma ואחרים), הכוללת את יתרונותיהם ומגבלותיהם הן מן ההיבט הטכנולוגי והן מן ההיבט של קוד פתוח (open source). כמו כן, הוצגו אפשרויות להתאמה ייעודית ולשימוש מקומי ללא תלות בשירותי ענן, במטרה לאפשר למורים לבחור את המודל המתאים ביותר לצורכיהם ההוראתיים והמחקריים.

במפגש הרביעי נלמדו שלושה מרכיבים מרכזיים בשילוב בינה מלאכותית בתהליך הפיתוח:

- כלים ותוספים לכתובת קוד מבוסס AI המספקים למפתחים סיוע אוטומטי ביצירת קטעי קוד מותאמים אישית בזמן אמת באמצעות ממשקי IDE ואתרי אינטרנט ייעודיים;
- יישום גישת RAG (Retrieval-Augmented Generation) לחיבור המערכת למאגרי ידע חיצוניים, מה שמבטיח זמינות מידע עדכני ורלוונטי בכל רגע נתון;

- עבודה עם מסמכי PDF וקבצים דומים הכוללת המרת תיעוד טקסטואלי למבני נתונים הניתנים לחיפוש ולשאילת שאלות חכמות, ובכך מאפשרת הפקת תובנות ישירות מתוך מסמכים מורכבים.

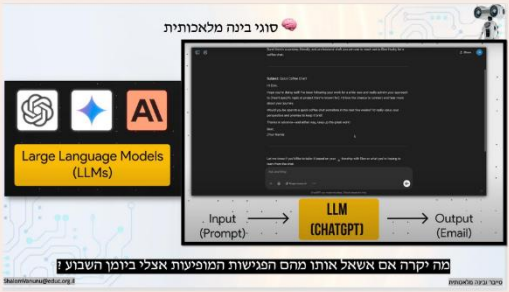
שילוב שלושה מרכיבים אלה יוצר תשתית הוליסטית לפיתוח תוכנה מהיר, מדויק ואינטואיטיבי יותר.

מפגש רביעי: סוכני AI וניהול ידע

- היכרות עם AI Agent (סוכני בינה מלאכותית) – הדרך לכתוב סוכני בינה מלאכותית בפיתוח עם כלים ותהליך ReAct
- צללו לעולם סוכני ה-AI תוכנות אוטונומיות שמסוגלות לבצע משימות מורכבות על בסיס הוראות, תוך כדי קבלת החלטות ותכנון צעדים. למדנו כיצד לבנות סוכנים כאלה ב-Python תוך שימוש בכלים ובתהליך ReAct (Reasoning and Acting) שאיפשר להם להגיב באופן דינמי.
- עבודה מול מאגרי מידע – RAG (Retrieval-Augmented Generation) היכולת לאחזר מידע רלוונטי ממקורות ידע חיצוניים:
- הבנו כיצד לחבר מודלי AI למאגרי מידע חיצוניים באמצעות טכניקת RAG. זה איפשר למודלים לאחזר מידע ספציפי ועדכני ממסמכים, מאגרי נתונים ואתרים, ולשלב אותו בתשובותיהם, מה שהעלה משמעותית את דיוקם ורלוונטיות המידע שסיפקו.
- עבודה מול מאגרי מידע PDF וכדומה:
- הדגמנו כיצד ליישם את עקרונות ה-RAG על מסמכי PDF ופורמטים דומים. למדנו כיצד לאפשר למודלי "AI ללקרוא" ולהבין את התוכן במסמכים אלו, לענות על שאלות ספציפיות מתוכם, ולחלץ מידע נחוץ ביעילות.

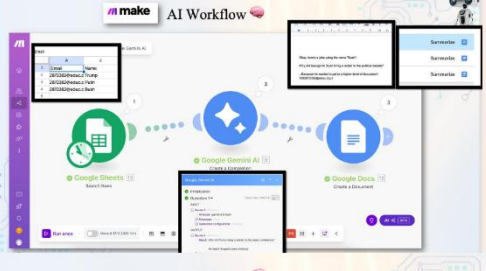


Made with GAMMA



AI Agent Python

מפגש רביעי: סוכני AI וניהול ידע



AI Agent

סוכני AI – (AI Agents) קבלת החלטות אוטונומית
 המהפכה: החלפת מקבל ההחלטות האנושי ב-LLM
 ההבדל המהותי בין AI workflow ל AI Agent, הוא שסוכן AI מחליף את מקבל ההחלטות האנושי ב-LLM המשמעות היא שהסוכן חייב להיות מסוגל לחשוב ולפעול באופן אוטונומי עם כלים שיש ברשותו.
 תכונות מפתח של סוכני AI: הסקה (Reason) סוכן ה-AI חייב להיות מסוגל לחשוב ולנתח את הדרך הטובה ביותר לחשוד מטרה מסוימת.
 "סוכן ה-AI חייב 'לחשוב' מהי הדרך היעילה ביותר..."

מטרות הלמידה

- לפתח Agent ב-Python המשלב "חשיבה-פעולה-משוב" (ReAct) ותומך בכלי עזר חיצוניים.
- לטעון מסמכי PDF, לחלץ מהם embedding ולאנדקס אותם לצורכי אחזור.
- ליישם RAG כך שה Agent יחזיר תשובות נתמכות מקור, כולל "source grounding".

- להבין שיקולי אבטחת מידע, הרשאות וגישה מבוקרת בעת חיבור מודלים למסדי ידע רגישים.
- לתכנן זרימת-עבודה אוטומטית: טריגר-משימה ← שלב Tool Use ← הפקת תשובה ← שלב Verification.

במפגש החמישי והאחרון הציגו המורים את הפרויקטים שהכינו בהתאם להנחיות מטלת הסיום בקורס "השתלמות הגנת סייבר בשילוב בינה מלאכותית", אשר דרשו הגשת פרויקט מסכם על פני שני עמודים לפחות הכולל תיאור ויזואלי וטקסטואלי של הארכיטקטורה, הסבר על מימוש הקוד לשילוב גישה בינה מלאכותית (למשל חיבור למודל Gemini API) יחד עם מאגר נתונים (SQLite או MongoDB), וכן דף רפלקציה אישי. במסגרת הפרויקט פותחו שרת ולקוח התקשורת מבוססי Socket בסגנון Server-Client (או P2P), כאשר מומלץ היה להדמות תרחישי סייבר אמיתיים כגון זיהוי תקיפה או ניהול הרשאות. בצד השרת הוטמעו קריאות ל-AI API לצורך עיבוד קלט, זיהוי תבניות ותגובה לאירועים, ואילו בצד הלקוח יועד ממשק גרפי אינטראקטיבי (למשל ב-Tkinter) להזנת נתונים, הצגת תוצאות ה-AI והדמיית תגובה לאירוע אבטחה. הקוד הוגש בתיקיית Server ובתיקיית Client נפרדות, ודף הרפלקציה סיפק תובנות על תהליך הלמידה, האתגרים והפתרונות שננקטו, כמתואר להלן:

הקורס "בינה מלאכותית בהתמחות סייבר" בהנחייתו של שלום ואנונו היווה עבורי חוויה לימודית מעשירה, חדשנית ומאתגרת מאוד. זהו אחד מהקורסים הטובים ביותר, שמצליחים לשלב בצורה הרמונית בין עקרונות מתקדמים לבין יישומים פרקטיים בעולם האמיתי של הגנת הסייבר. כבר מהמפגש הראשון ניתן היה להבחין ברמה הגבוהה של ההוראה, במחויבות של המרצה וביכולת שלו להנגיש נושאים מורכבים בצורה ברורה, מעשית וסוחפת. לסיכום, הקורס הזה היווה עבורי שילוב נדיר בין תיאוריה לעשייה. אני יוצא ממנו עם ידע מעשי, ביטחון טכני רב יותר, ובעיקר עם השראה להמשיך לחקור, ללמוד ולפתח חומרי לימוד בתחומי ה-AI והסייבר. אני ממליץ עליו בחום לכל מי שמבקש להעמיק בעולם מרתק זה טכנולוגיים – ובעיקר למי שזוכה ללמוד אותו בהנחייתו של שלום ואנונו.
ברצוני לציין במיוחד את שלום, שהנחה את ההשתלמות בגישה מקצועית, סבלנית ומדויקת. שלום שמר על קו הדרכה ממוקד ואחיד, לא הוסח על ידי הערות שוליים, והצליח לייצר שיעורים הרמוניים, ממוקדים וזורמים. התמיכה וההבהרות שניתנו – תמיד הגיעו לאחר ההקניה, באופן ששמר על רצף תקין של ההוראה. הסדר והארגון ניכרו בכל שלב – החל מהלוח המדויק, דרך החומרים, התזכורות, אוסף הקישורים ועד להעשרה המקצועית שנלוותה לאורך כל הדרך. מעל הכול בלטה תחושת נתינה, ענווה ומקצועיות עמוקה. השתלמות זו הצליחה לשלב בין תכנים רלוונטיים ומעמיקים לבין מסגרת ניהולית מסודרת ונעימה. היה מעולה – שלום -תודה רבה!
מיני פרויקט זה מדגים כיצד ניתן לשלב בין עולם הסייבר לעולם הבינה המלאכותית באמצעות כלים פשוטים ונגישים: ממשק גרפי נוח socket בין לקוח לשרת שימוש ב-Gemini API מסד נתונים לשמירת המידע

נהניתי לבצע את המטלה, ומאוד נהניתי ללמוד את ההשתלמות הזו ולהיחשף לאפשרות לשלב API AI בצד השרת.
אני מתכוון לשלב את זה בהוראת פרויקט הגמר בכיתה י"ב ויש לי כבר רעיונות נוספים.
תודה למרצה שלום ואנוני, על ההוראה ועל החומרים המעניינים!

סיכום

ההשתלמות מגלמת נקודת מפנה מהותית בהטמעת כלים מתקדמים של בינה מלאכותית בפרויקטים בהנדסת תוכנה בחלופת סייבר, שכן היא מאפשרת למורים ולסטודנטים להרחיב את מסגרת העבודה הקלאסית ולשלב יכולות ניתוח חכם, זיהוי תבניות ותגובה אוטומטית לאיומים באמצעות חיבור לפרוטוקול Gemini API. בכך היא פותחת אופק חדש של פיתוח מערכות הגנה מתקדמות, המשלבות הן את ההבנה התיאורטית של אבטחת מידע והן את היישום הפרקטי של מנועי AI בזמן אמת, מקדמת גישה היברידית המשצבת את הפדגוגיה סביב חשיבה עניינית-טכנולוגית, מחזקת את הקשר בין התיאוריה לתרגול ומעניקה למשתתפים כלים ליצירת פתרונות סייבר חדשניים המותאמים לאתגרים העדכניים ביותר.

מילון מושגים - השתלמות בינה מלאכותית למפתחים

מושגי יסוד בבינה מלאכותית

AI Artificial Intelligence - בינה מלאכותית: טכנולוגיה המסוגלת לבצע משימות הדורשות אינטליגנציה אנושית, כגון הבנת שפה, זיהוי דפוסים ופתרון בעיות מורכבות.

LLM - Large Language Model מודל שפה גדול: מודל בינה מלאכותית המאומן על כמויות עצומות של טקסט, המסוגל להבין ולייצר טקסט באיכות גבוהה בשפה טבעית.

Local LLM - מודל שפה גדול מקומי: מודל AI הרץ על המחשב האישי ללא תלות בחיבור לאינטרנט או בשירותי ענן חיצוניים, המאפשר עבודה פרטית ומהירה.

כלי פיתוח וממשקים

API - Application Programming Interface - ממשק תכנות יישומים: ממשק המאפשר תקשורת ואינטראקציה בין מערכות תוכנה שונות, כגון חיבור האפליקציה של המשתמש לשירותי AI של גוגל.

Google API: ממשק התכנות של גוגל המאפשר גישה לשירותי הבינה המלאכותית של החברה, כמו Gemini.

CLI - Command Line Interface - ממשק שורת פקודה: כלי לאינטראקציה ישירה עם המחשב או שירותים מרוחקים באמצעות פקודות טקסט, ללא צורך בממשק גרפי.

Python: שפת תכנות פופולרית ונגישה המשמשת רבות לפיתוח יישומי בינה מלאכותית, עיבוד נתונים ואוטומציה.

טכניקות עבודה עם AI

Prompt פרומפט: הוראה, שאלה או הנחיה שנשלחת למודל בינה מלאכותית כדי לקבל תגובה או פלט רצוי ומדויק.

Example Prompt - פרומפט לדוגמה: שימוש בדוגמאות קונקרטיות בתוך הפרומפט כדי לכוון את המודל להבין בדיוק איזה סוג תגובה מצופה ממנו.

Continues Prompt - פרומפט מתמשך: טכניקה לשמירה על הקשר והקונטקסט במהלך שיחה ארוכה עם המודל, כך שהוא "זוכר" מה נאמר קודם.

Structure - מבנה פרומפט: ארגון נכון ומתודי של הפרומפט, הכולל הקשר, הוראות ברורות ודוגמאות, לקבלת תוצאות מדויקות ואיכותיות.

מושגים מתקדמים

AI Agent - סוכן בינה מלאכותית: תוכנה אוטונומית המסוגלת לבצע משימות מורכבות באופן עצמאי, תוך קבלת החלטות, תכנון צעדים ושימוש בכלים שונים.

ReAct - Reasoning and Acting - הגיון ופעולה: תהליך או מתודולוגיה המאפשרת לסוכני בינה מלאכותית להגיב באופן דינמי ואינטליגנטי, תוך שילוב של חשיבה לוגית ופעולה מעשית.

RAG - Retrieval-Augmented Generation - יצירה מתוגברת אחזור: טכניקה מתקדמת לחיבור מודלי בינה מלאכותית למאגרי מידע חיצוניים, המאפשרת למודל לאחזר מידע רלוונטי ועדכני ולשלב בתשובותיו.

Vibe Coding: קונספט מהפכני של פיתוח אפליקציות Full Stack באמצעות אינטראקציה עם בינה מלאכותית דרך פרומפטים בלבד, ללא צורך בכתיבת קוד רב באופן ידני.

פלטפורמות וכלים

Ollama: פלטפורמה המאפשרת הקמה והרצה פשוטה של מודלי שפה גדולים על המחשב האישי, ללא תלות בשירותי ענן.

Open Source - קוד פתוח: תוכנה הזמינה לשימוש, שינוי והפצה חופשית, המאפשרת גמישות, שקיפות ויכולת התאמה אישית.

פורמטים ומבני נתונים

JSON (JavaScript Object Notation): פורמט קל ונפוץ להעברת נתונים בין מערכות בצורה מובנית וקריאה, המשמש רבות בתקשורת עם API.

Full Stack - פיתוח מלא: גישת פיתוח המכסה את כל שכבות האפליקציה - ממשק המשתמש (חזית), לוגיקת השרת (עורף) ומסד הנתונים.