

კურსის სახელწოდება: AI ციფრული სახელოსნო: შექმენი შენი პირველი ჩატბოტი

აუდიტორია: X-XII კლასის მოსწავლეები (პერსონა: 17 წლის ლუკა)

ხანგრძლივობა: 14 შეხვედრა

ფორმატი: 14 პრაქტიკული დავალება (13 AnITa-ს Python-ის სიმულატორში და 1 ფინალური პროექტი Google Colab-ში).

შეხვედრა 1: მოგზაურობა ხელოვნური ინტელექტის სამყაროში

1. რა არის ხელოვნური ინტელექტი (AI)?
 - 1.1. AI-ს მარტივი განმარტება: როგორ ვასწავლოთ კომპიუტერს "ფიქრი".
 - 1.2. ცხოვრებისეული მაგალითები: ხელოვნური ინტელექტი შენს სმარტფონში, Google Map-ში და YouTube-ში.
 - 1.3. კურსის მთავარი მიზანი: ჩვენ შევექმნით საკუთარ "მოაზროვნე" პროგრამას - ჩატბოტს.
2. როგორ "ფიქრობს" კომპიუტერი?
3.
 - 2.1. განსხვავება ადამიანისა და მანქანის ინტელექტს შორის.
 - 2.2. ხელოვნური ინტელექტის ტიპები: ვიწრო (ANI), ზოგადი (AGI) და სუპერინტელექტი (ASI).
 - 2.3. ტიურინგის ტესტი: როდის გაუტოლდება კომპიუტერის ინტელექტი ადამიანის ინტელექტს?
4. ხელოვნური ინტელექტის გავლენა ჩვენს მომავალზე.
 - 3.1. მომავლის პროფესიები და ხელოვნური ინტელექტის უნარ-ჩვევების საჭიროება.
 - 3.2. ეთიკური დილემები: ვისი ბრალია, როცა ხელოვნური ინტელექტი შეცდომას უშვებს?
 - 3.3. ანITa-ს პლატფორმისა და Python-ის სიმულატორის გაცნობა.

პრაქტიკული დავალება 1: ჩემი პირველი პროგრამა

- **აღწერა:** დაწერე Python კოდი, რომელიც ეკრანზე დაბეჭდავს მისალმებას შენი მომავალი ჩატბოტის სახელით. მაგალითად: "გამარჯობა, მე ვარ AnITa-ს ასისტენტი! მზად ვარ სწავლისთვის!".
- **პლატფორმა:** AnITa-ს Python-ის სიმულატორი.

შეხვედრა 2: პირველი ნაბიჯები Python-ში – ჩვენი ჩატბოტის "მეხსიერება"

1. რატომ Python-ი?

1.1. Python-ი, როგორც ადამიანურ ენასთან ყველაზე ახლოს მდგომი პროგრამირების ენა.

1.2. რატომ ირჩევენ Python-ს ხელოვნურ ინტელექტთან სამუშაოდ.

2. ცვლადები და მონაცემთა ტიპები.

2.1. ცვლადი, როგორც ინფორმაციის შესანახი "ყუთი" (სახელი, ასაკი, ქალაქი).

2.2. ძირითადი მონაცემთა ტიპები: ტექსტი (str), მთელი რიცხვი (int), ათწილადი (float), ლოგიკური (bool).

2.3. print() ფუნქცია: როგორ "ავალაპარაკოთ" ჩვენი პროგრამა.

3. მონაცემების დინამიურად გამოყენება.

3.1. ფორმატირებული სტრიქონები (f-strings): როგორ ჩავსვათ ცვლადები ტექსტში.

3.2. კომენტარების (#) როლი: შენიშვნები საკუთარი თავისთვის და სხვებისთვის.

პრაქტიკული დავალება 2: პერსონალური მისაღება

- **აღწერა:** შექმენი ორი ცვლადი: `bot_name` და `user_name`. პროგრამამ უნდა გამოიყენოს ეს ცვლადები და დაბეჭდოს პერსონალური მისაღება, მაგალითად: გამარჯობა, [მომხმარებლის სახელი]! მე ვარ [ბოტის სახელი].
- **პლატფორმა:** AnITa-ს Python-ის სიმულატორი.

შეხვედრა 3: გადანყვეტილებების მიღება კოდით – ვასწავლოთ ბოტს პასუხის გაცემა

1. მომხმარებლისგან ინფორმაციის მიღება.

1.1. input() ფუნქცია: როგორ დავუშვათ მომხმარებელს კითხვა და მივიღოთ პასუხი.

1.2. მიღებული ინფორმაციის შენახვა ცვლადში.

2. პირობითი ოპერატორები: if-else.

2.1. ანალოგია: თუ ამინდი კარგია, წავალ გარეთ, სხვა შემთხვევაში - დავრჩები სახლში.

2.2. შედარების ოპერატორები (==, !=, <, >): როგორ შევადაროთ მნიშვნელობები.

2.3. if, elif, else კონსტრუქცია ლოგიკური განშტოებების შესაქმნელად.

3. მარტივი დიალოგის აგება.

3.1. როგორ დავწეროთ პროგრამა, რომელიც მომხმარებლის პასუხის მიხედვით განსხვავებულად მოიქცევა.

3.2. ფსევდოკოდი: ლოგიკის დაგეგმვა კოდის დაწერამდე.

პრაქტიკული დავალება 3: "მარტივი დიალოგი"

- **აღწერა:** დაწერე პროგრამა, რომელიც მომხმარებელს ჰკითხავს "როგორ ხარ?". თუ მომხმარებელი დაწერს "კარგად", ბოტმა უნდა უპასუხოს "მიხარია!". ყველა სხვა შემთხვევაში, ბოტმა უნდა უპასუხოს "იმედია, ყველაფერი კარგად იქნება!".
 - **პლატფორმა:** AnITa-ს Python-ის სიმულატორი.
-

შეხვედრა 4: მონაცემთა სტრუქტურები (სიები) და ციკლები – გავამრავალფეროვნოთ ბოტის პასუხები

1. მონაცემთა სტრუქტურები: სიები (Lists).
 - 1.1. სია, როგორც ერთი ცვლადის ქვეშ გაერთიანებული ბევრი მნიშვნელობა.
 - 1.2. სიის შექმნა, ელემენტების დამატება (append) და წაშლა (remove).
 - 1.3. ელემენტზე წვდომა ინდექსის ([0], [1]) საშუალებით.
2. მოქმედებების გამოვრება: for ციკლი.
 - 2.1. ანალოგია: როგორ დავათვალიეროთ სავაჭრო სიის ყველა პროდუქტი სათითაოდ.
 - 2.2. for ციკლის გამოყენება სიის ყველა ელემენტზე მოქმედების შესასრულებლად.
3. შემთხვევითობის ელემენტი.
 - 3.1. random ბიბლიოთეკის იმპორტი.
 - 3.2. random.choice() ფუნქცია სიიდან შემთხვევითი ელემენტის ამოსარჩევად.

პრაქტიკული დავალება 4: "არაპროგნოზირებადი ბოტი"

- **აღწერა:** შექმენი მისაღმებების სია (მაგ. ["გამარჯობა!", "მოგესალმები!", "სალამი!"]). დაწერე კოდი, რომელიც random.choice()-ის გამოყენებით, პროგრამის ყოველი გაშვებისას შემთხვევით მისაღმებას დაბეჭდავს.
 - **პლატფორმა:** AnITa-ს Python-ის სიმულატორი.
-

შეხვედრა 5: მონაცემთა სტრუქტურები (ლექსიკონები) – ვქმნით ბოტის ტვინს

1. მონაცემთა სტრუქტურები: ლექსიკონები (Dictionaries).
 - 1.1. ლექსიკონი, როგორც წყვილების საცავი: გასაღები-მნიშვნელობა.

1.2. ანალოგია: სატელეფონო წიგნაკი (სახელი - ნომერი) ან განმარტებითი ლექსიკონი (სიტყვა - განმარტება). 1.3. ლექსიკონის შექმნა, ახალი წყვილის დამატება და მნიშვნელობაზე წვდომა გასაღებით.

2. ჩატბოტის მოდელი: მიზანი-პასუხი

2.1. მიზანი (Intent): რა მიზანი აქვს მომხმარებელს, რისი თქმა სურს

მომხმარებელს (მაგ. `greeting`, `goodbye`).

2.2. პასუხი (Response): რა პასუხი უნდა გასცეს ბოტმა მომხმარებლის შეკითხვას?

2.3. ლექსიკონის გამოყენება ამ მოდელის ასაგებად: `{'greeting': 'გამარჯობა!', 'goodbye': 'ნახვამდის!'}`.

3. ლექსიკონისა და `if` პირობის კომბინირება.

3.1. როგორ შევამოწმოთ, არსებობს თუ არა მომხმარებლის შეტყობინება ჩვენი ლექსიკონის გასაღებებში.

3.2. `.get()` მეთოდი უსაფრთხო პასუხის მისაღებად, თუ გასაღები არ მოიძებნა.

პრაქტიკული დავალება 5: ლექსიკონის პრინციპით შექმნილი ბოტი

- **აღწერა:** შექმენი ლექსიკონი, სადაც გასაღები იქნება მომხმარებლის სავარაუდო შეტყობინება ("გამარჯობა", "როგორ ხარ", "ნახვამდის"), ხოლო მნიშვნელობა - ბოტის პასუხი. დაწერე კოდი, რომელიც მომხმარებლის შეტყობინების მიხედვით შესაბამის პასუხს დაბეჭდავს ლექსიკონიდან.
- **პლატფორმა:** AnITa-ს Python-ის სიმულატორი.

შეხვედრა 6: ფუნქციები: ვაქციოთ ჩვენი კოდი მრავალჯერადად გამოყენებად ინსტრუმენტად

1. რა არის ფუნქცია?

1.1. ფუნქცია, როგორც კოდის ბლოკი, რომელიც კონკრეტულ ამოცანას ასრულებს.

1.2. ანალოგია: კალკულატორზე ლილაკი `+`, რომელიც ყოველთვის შეკრების ოპერაციას ასრულებს.

1.3. `def` საკვანძო სიტყვა ფუნქციის შესაქმნელად.

2. ფუნქციის პარამეტრები და დაბრუნებული მნიშვნელობა.

2.1. პარამეტრი (არგუმენტი): ინფორმაცია, რომელსაც ფუნქციას გადავცემთ სამუშაოდ.

2.2. `return` საკვანძო სიტყვა: შედეგის დაბრუნება ფუნქციიდან.

3. ჩატბოტის ლოგიკის ფუნქციაში მოქცევა.

3.1. შევქმნათ ფუნქცია `get_response(user_input)`, რომელიც მიიღებს მომხმარებლის ტექსტს.

3.2. ფუნქციის შიგნით განვათავსოთ ლექსიკონთან და `if-else` ლოგიკასთან

დაკავშირებული კოდი.

3.3. ფუნქციამ უნდა დააბრუნოს (return) ბოტის საბოლოო პასუხი.

პრაქტიკული დავალება 6: ფუნქციონალური ბოტი

- **აღწერა:** წინა დავალების კოდი გადააკეთე ისე, რომ ბოტის პასუხის გენერირების ლოგიკა მოექცეს ფუნქციაში `get_bot_response(message)`. პროგრამამ უნდა გამოიძახოს ეს ფუნქცია და დაბეჭდოს მის მიერ დაბრუნებული პასუხი.
 - **პლატფორმა:** AnITa-ს Python-ის სიმულატორი.
-

შეხვედრა 7: შესავალი ბუნებრივი ენის დამუშავებაში (NLP) – ვასწავლოთ ბოტს წაკითხვა

1. როგორ ესმით კომპიუტერებს ადამიანური ენა?
 - 1.1. NLP-ის (Natural Language Processing) განმარტება.
 - 1.2. ძირითადი გამოწვევა: ადამიანური ენა სავსეა ორაზროვნებითა და კონტექსტით.
2. ტექსტის მომზადება ანალიზისთვის.
 - 2.1. ტექსტის დაპატარავება (`.lower()`): "გამარჯობა" და "გაგიმარჯოთ" ერთი და იგივეა.
 - 2.2. ტოკენიზაცია: ტექსტის დაშლა სიტყვებად (ტოკენებად).
 - 2.3. არასაჭირო სიტყვების (Stop Words) და სასვენი ნიშნების მოშორება.
3. სიტყვების ფუძეების პოვნა.
 - 3.1. სტემინგი და ლემატიზაცია: რატომ არის "წავიდა", "მიდის", "წასვლა" ერთი და იგივე ძირის მქონე.
 - 3.2. როგორ გვეხმარება ეს ბოტისთვის მომხმარებლის განზრახვის უკეთ გაგებაში.

პრაქტიკული დავალება 7: ტექსტის გამზადება

- **აღწერა:** დაწერე ფუნქცია `clean_text(text)`, რომელიც მიიღებს წინადადებას, მოაშორებს სასვენი ნიშნებს (., !, ?) და დააბრუნებს გასუფთავებულ ტექსტს.
 - **პლატფორმა:** AnITa-ს Python-ის სიმულატორი.
-

შეხვედრა 8: შესავალი მანქანური სწავლებაში (Machine Learning)

1. რა არის მანქანური სწავლება?

- 1.1. განსხვავება ტრადიციულ პროგრამირებასა (წესების გაწერა) და ML-ს (მონაცემებიდან სწავლა) შორის.
- 1.2. ანალოგია: ბავშვი სწავლობს ცხოველების ცნობას სურათებიდან და არა მათი აღწერილობის წაკითხვით.

2. მანქანური სწავლების ტიპები.

- 2.1. სწავლა მასწავლებლის დახმარებით (Supervised Learning): სწავლა "მასწავლებელთან" ერთად (დასათაურებული მონაცემები).
- 2.2. დამოუკიდებლად სწავლა (Unsupervised Learning): ფარული კანონზომიერებების (Patterns) პოვნა "მასწავლებლის" გარეშე.
- 2.3. ცდისა და შეცდომის მეთოდით სწავლა (Reinforcement Learning): სწავლა ცდისა და შეცდომის მეთოდით.

3. მონაცემების როლი.

- 3.1. სავარჯიშო (Training) და სატესტო (Test) მონაცემების მნიშვნელობა.
- 3.2. "ნაგავი შედის, ნაგავი გამოდის" (Garbage In, Garbage Out) პრინციპი.
- 3.3. მონაცემთა მიკერძოების (Bias) პრობლემა და მისი საფრთხეები.

პრაქტიკული დავალება 8: "პრობლემის იდენტიფიკაცია კოდით"

- **აღწერა:** შექმენი Python ლექსიკონი, სადაც გასაღები იქნება პრობლემის აღწერა (მაგ. "სახლის ფასის პროგნოზირება"), ხოლო მნიშვნელობა - მანქანური სწავლების შესაბამისი ტიპი. დაწერე კოდი, რომელიც for ციკლის გამოყენებით დაბეჭდავს თითოეულ პრობლემას და მის ტიპს.
- **პლატფორმა:** AnITa-ს Python-ის სიმულატორი.

შეხვედრა 9: გელამხედველობითი სწავლება განზრახვის კლასიფიკაცია

1. რა არის კლასიფიკაცია?

- 1.1. პრობლემის ფორმულირება: ობიექტის კატეგორიის (კლასის) განსაზღვრა.
- 1.2. მაგალითები: სპამის ფილტრი, სურათზე კატის ან ძაღლის ამოცნობა.
- 1.3. ჩვენი ამოცანა: მომხმარებლის შეტყობინების კლასიფიკაცია განზრახვის მიხედვით (greeting, question, goodbye).

2. მონაცემების მომზადება კლასიფიკაციისთვის.

- 2.1. ნიშან-თვისებების (Features) და სამიზნე ცვლადის (Label/Target) ცნება.
- 2.2. როგორ ვაქციოთ ტექსტი რიცხვებად, რომლებიც კომპიუტერს ესმის (Bag-of-Words მოდელი).

3. პირველი მოდელის იდეა: გადაწყვეტილების ხე.
 - 3.1. როგორ მუშაობს ალგორითმი, რომელიც მონაცემების გასაყოფად ლოგიკურ "კი/არა" კითხვებს სვამს.
 - 3.2. ანალოგია: თამაში "გამოიცანი ვინ ვარ?".

პრაქტიკული დავალება 9: "მონაცემთა ბაზის შექმნა"

- **აღწერა:** შექმენი ორი სია. პირველ სიაში ჩანერე 3-4 წინადადება, ხოლო მეორეში - მათი შესაბამისი განზრახვები (მაგ. "მისაღმება", "კითხვა"). დაწერე კოდი, რომელიც for ციკლის გამოყენებით ეკრანზე გამოიტანს თითოეულ წინადადებას და მის განზრახვას.
 - **პლატფორმა:** AnITa-ს Python-ის სიმულატორი.
-

შეხვედრა 10: მონაცემთა ანალიზის ბიბლიოთეკები: Pandas

1. რა არის ბიბლიოთეკა პროგრამირებაში?
 - 1.1. მზა ინსტრუმენტების გამოყენების უპირატესობა.
 - 1.2. `import pandas as pd` - როგორ "გამოვიძახოთ" პროფესიონალი მონაცემთა ანალიტიკოსი.
2. Pandas ბიბლიოთეკა.
 - 2.1. `DataFrame` - მონაცემების წარმოდგენა ჭკვიანი ცხრილის სახით.
 - 2.2. მონაცემების ჩატვირთვა CSV ფაილიდან.
 - 2.3. მონაცემების დათვალიერება: `head()`, `describe()`, `info()`.
3. მონაცემებთან მუშაობა.
 - 3.1. კონკრეტულ სვეტებსა და რიგებზე წვდომა.
 - 3.2. მონაცემების ფილტრაცია და საბაზისო ანალიზი.

პრაქტიკული დავალება 10: მონაცემების დათვალიერება

- **აღწერა:** გამოიყენე Pandas, რათა ჩატვირთო CSV ფაილი, რომელიც შეიცავს 2 სვეტს: `text` და `intent`. დაბეჭდე:
 1. მონაცემთა ბაზის პირველი 5 რიგი.
 - 2. თითოეული განზრახვის (`intent`) რაოდენობა მონაცემთა ბაზაში.
 - **პლატფორმა:** AnITa-ს Python-ის სიმულატორი.
-

შეხვედრა 11: პირველი ML მოდელის აგება: Scikit-learn

1. **Scikit-learn ბიბლიოთეკა.**
 - 1.1. მრავალფუნქციური ინსტრუმენტი მანქანური სწავლებისთვის Python-ში.
 - 1.2. მოდელის შექმნის 4 ძირითადი ეტაპი: იმპორტი, ინიციალიზაცია, ვარჯიში, პროგნოზირება.
2. ტექსტის ვექტორიზაცია.
 - 2.1. **CountVectorizer**: ტექსტების გადაქცევა რიცხვების ვექტორებად. 2.2. როგორ იქმნება "ყველა სიტყვის ლექსიკონი" და როგორ ხდება სიტყვების სიხშირის გამოთვლა.
3. მოდელის აგება და ვარჯიში.
 - 3.1. ლოგისტიკური რეგრესიის (Logistic Regression) მოდელის გამოყენება კლასიფიკაციისთვის.
 - 3.2. `model.fit(X_train, y_train)`: მოდელის დატრენინგების პროცესი.
 - 3.3. `model.predict(X_test)`: ახალი მონაცემებისთვის პროგნოზის გაკეთება.

პრაქტიკული დავალება 11: "განზრახვის პროგნოზირება"

- **აღწერა:** წინა დავალების მონაცემებზე (text, intent) დაყრდნობით, **scikit-learn**-ის გამოყენებით, გაავარჯიშე **CountVectorizer** და **LogisticRegression** მოდელი. გამოიყენე გაწვრთნილი მოდელი, რათა იწინასწარმეტყველო ახალი წინადადების ("დღეს კარგი ამინდია") განზრახვა.
- **პლატფორმა:** AnITa-ს Python-ის სიმულატორი.=

შეხვედრა 12: შესავალი ნეირონულ ქსელებსა და ღრმა სწავლებაში (Deep Learning)

1. რა არის ნეირონული ქსელი?
 - 1.1. ადამიანის ტვინის სტრუქტურით შთაგონებული მოდელი.
 - 1.2. ანალოგია: კომპანია, სადაც სხვადასხვა დეპარტამენტი (ფენა) და თანამშრომელი (ნეირონი) ერთად მუშაობენ რთული პრობლემის გადასაჭრელად.
 - 1.3. ფენების, ნეირონებისა და კავშირების კონცეფცია.
2. რა არის ღრმა სწავლება (Deep Learning)?
 - 2.1. მრავალფენიანი ღრმა ნეირონული ქსელები.
 - 2.2. რატომ არის Deep Learning განსაკუთრებით ძლიერი ენასთან და სურათებთან მუშაობისას.
3. დიდი ენობრივი მოდელები (LLMs).
 - 3.1. რა არის Transformer არქიტექტურა და მისი გარდამტეხი როლი

ხელოვნური ინტელექტის ევოლუციაში.

3.2. როგორ მუშაობენ ChatGPT, Gemini და სხვა თანამედროვე მოდელები.

3.3. Fine-tuning კონცეფცია: როგორ ვასწავლოთ გენიოსს ახალი, კონკრეტული საგანი.

პრაქტიკული დავალება 12: ნეირონული ქსელის აღწერა კოდით

- **აღწერა:** შექმენი ლექსიკონი, რომელიც სტრუქტურულად აღწერს ნეირონულ ქსელს (მაგ. `{'input_layer': 784, 'hidden_layer_1': 128, 'output_layer': 10}`). დაწერე კოდი, რომელიც ამ ლექსიკონის მიხედვით დაბეჭდავს ქსელის სტრუქტურის აღწერას.
- **პლატფორმა:** AnITa-ს Python-ის სიმულატორი.

შეხვედრა 13: ფინალური პროექტისთვის მზადება: Google Colab

1. რა არის Google Colaboratory (Colab)?
 - 1.1. უფასო Jupyter Notebook გარემო ღრუბელში, Google-ის სერვერებზე.
 - 1.2. რატომ არის ის იდეალური ხელსაწყო მანქანური სწავლების პროექტებისთვის.
 - 1.3. Colab-ის ინტერფეისის გაცნობა: ტექსტური და კოდის უჯრები.
2. Colab-ში მუშაობის საფუძვლები.
 - 2.1. როგორ გავხსნათ, შევინახოთ და გავაზიაროთ პროექტი.
 - 2.2. ბიბლიოთეკების ინსტალაცია `!pip install` ბრძანებით.
 - 2.3. ფაილების ატვირთვა და მათთან მუშაობა.
3. ფინალური პროექტის მიმოხილვა.
 - 3.1. პროექტის მიზანი: შევქმნათ მარტივი ჩატბოტი, რომელიც უპასუხებს კითხვებს AnITa-ს კურსის შესახებ.
 - 3.2. მონაცემთა ბაზის მომზადება: კითხვა-პასუხის წყვილების შექმნა.
 - 3.3. პროექტის შესრულების ეტაპების განხილვა.

პრაქტიკული დავალება 13: გავარჯიშება სიმულატორში

- **აღწერა:** AnITa-ს სიმულატორში, `pandas` ბიბლიოთეკის გამოყენებით შექმენი მარტივი DataFrame-ი, რომელიც შეიცავს 2 სვეტს (კითხვა, პასუხი) და 3 რიგს. დაბეჭდე ეს DataFrame-ი. ეს დაგეხმარება ფინალური პროექტის მონაცემთა ბაზის სტრუქტურის გააზრებაში.
- **პლატფორმა:** AnITa-ს Python-ის სიმულატორი.

შეხვედრა 14: მოდულის შეჯამება და ფინალური პროექტი

- რა ვისწავლეთ ამ კურსში?
 - გავიარეთ გზა `print("Hello World")`-დან მანქანური სწავლების მოდელის აგებამდე.
 - Python-ის საფუძვლები, NLP-ის კონცეფციები და ML-ის პრაქტიკული გამოყენება.
 - როგორ ვაქციოთ იდეა მოქმედ პროგრამად.
- ეთიკა და ხელოვნური ინტელექტი (შეჯამება).
 - მონაცემთა კონფიდენციალურობა და ალგორითმული მიკერძოება.
 - AI-ს პასუხისმგებლიანი გამოყენების მნიშვნელობა.
- ფინალურ პროექტზე მუშაობის დანწყება.
 - კითხვა-პასუხის სესია და პროექტთან დაკავშირებული საკითხების გარკვევა.
 - დამოუკიდებელი მუშაობის დანწყება მენტორის დახმარებით.

პრაქტიკული დავალება 14 (ფინალური პროექტი): "AnITa-ს ასისტენტი"

- აღწერა:** Google Colab-ში გამოიყენე კურსის განმავლობაში მიღებული ცოდნა და შექმენი მარტივი ჩატბოტი.
 - მონაცემები:** შექმენი მცირე ზომის კითხვა-პასუხის ბაზა AnITa-ს კურსის შესახებ (მაგ. "რამდენი შეხვედრაა კურსში?", "რას ნიშნავს NLP?").
 - მოდელი:** გამოიყენე `scikit-learn`-ი, რათა გაავარჯიშო კლასიფიკატორი, რომელიც მომხმარებლის კითხვის მიხედვით ამოიცნობს სწორ პასუხს (შეგიძლია გამოიყენო კოსინუსური მსგავსება ან სხვა მარტივი მეთოდი).
 - ინტერფეისი:** შექმენი ციკლი, რომელიც მომხმარებელს საშუალებას მისცემს, მუდმივად დაუსვას კითხვები ბოტს და მიიღოს მისგან პასუხები.
- პლატფორმა:** Google Colab (პროექტის ბმული უნდა აიტვირთოს AnITa-ს პლატფორმაზე).