

**Пловдивски университет "Паисий Хилендарски"  
Факултет по Математика и Информатика**

**Курсов проект по Въведение в изграждането на инфраструктури за обработката на данни с Python  
  
Тема:** Чатбот за https://fmi-plovdiv.org/, персонален асистен

**Изготвили:**Любомир Стойчев - 2301727002  
Габриела Балабанска - 2301727003  
Денис Троничев - 2301747002  
  
**Ръководител:**  
Ангел Георгиев  
  
**Дата:**  
16.01.2025г.

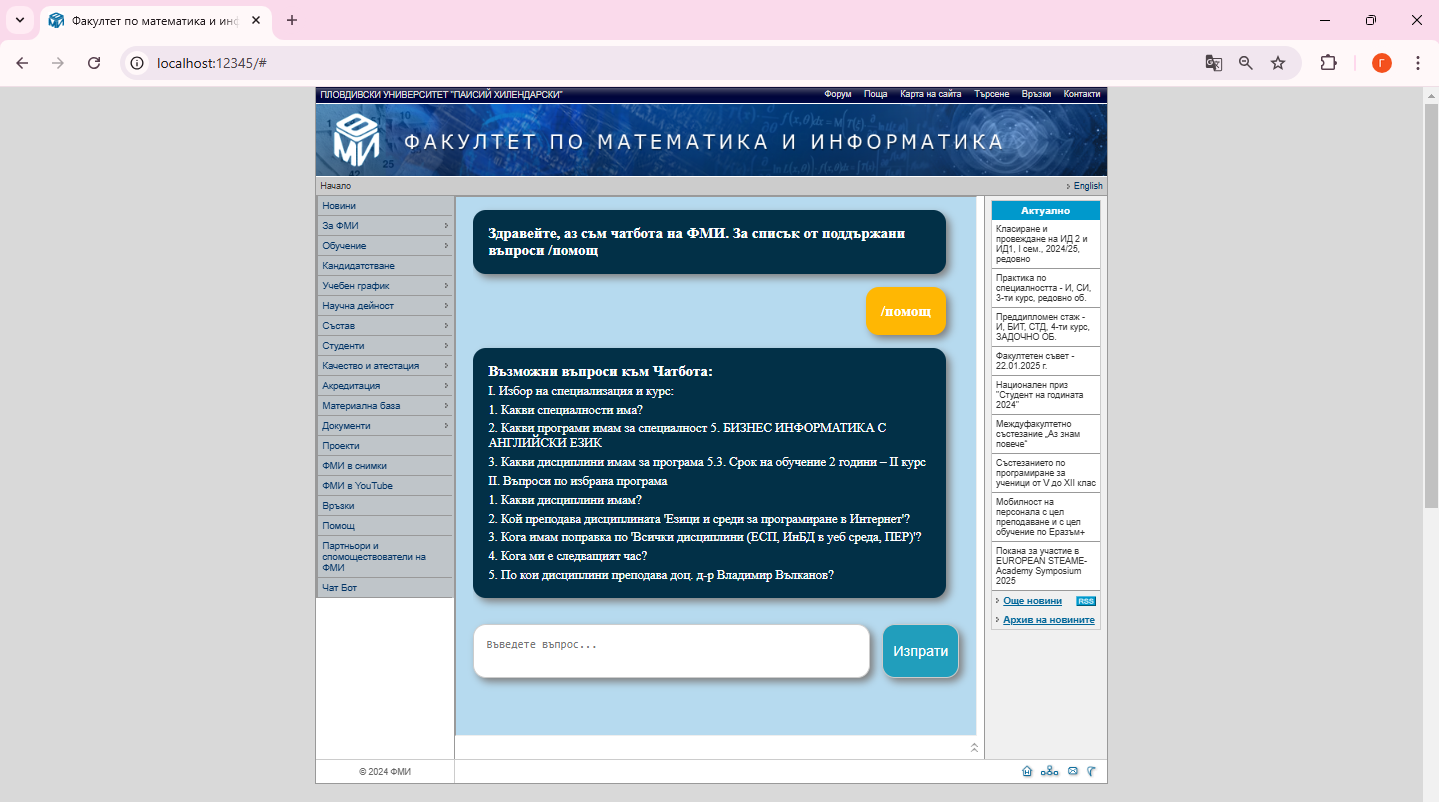
1. **Увод**

Проектът представлява Чатбот, интегриран в уебсайта на Пловдивския университет "Паисий Хилендарски" чрез HTTP Proxy. За реализацията на проекта са използвани Back-end на Python и Front-end на Angular 19.   
Чатботът има за цел да улесни студентите в магистърски курс като предоставя лесен достъп до информация относно магистърските програми, курсовете и семестрите във Факултета по математика и информатика.

**Принцип на работа на Чатбота**

1. **Скрейпинг на данни**  
   Чатботът започва с извличане на информация от официалния уебсайт на Пловдивския университет, използвайки техники за уеб скрейпинг. Това включва събиране на данни относно магистърските програми, курсовете, специалностите и семестрите, които са налични на сайта.
2. **Филтриране, преработка и корекция на данни**  
   След като данните бъдат събрани, те преминават през етап на филтриране и преработка. Това включва премахване на ненужни или неправилно структурирани елементи, както и коригиране на неясноти или неточности в събраната информация, за да бъде представена по ясен и достъпен начин.
3. **Използване на контролер за обработка на потребителски въпроси**  
   След като данните са готови, те се предават на контролер, който е обучен да разпознава различни типове потребителски въпроси и да предоставя точни и релевантни отговори. Контролерът обработва входните запитвания и ги свързва с подходящата информация от базата данни.
4. **Предоставяне на входни точки за Front-end**  
   Контролерът осигурява набор от API входни точки, които да могат да бъдат използвани от Front-end приложението. Тези входни точки позволяват комуникация между чатбота и потребителския интерфейс, осигурявайки правилното взаимодействие между backend и frontend слоевете на приложението.
5. **Комуникация с потребителя чрез Front-end**  
   Front-end частта на приложението осигурява удобен и интуитивен интерфейс за потребителите, който им позволява да взаимодействат с чатбота. Потребителите могат да задават въпроси, да избират специалности и курсове, а чатботът ще отговори на техните запитвания чрез динамично генериран текст в реално време.

Потребителите могат да избират специалност, след което да разглеждат наличните курсове и семестри, като накрая имат възможността да задават въпроси относно специфични аспекти на програмата.   
На Фигура 1 е показан сайта на университета, в който посредством HTTP Proxy е добавен Чатбота. При първоначално влизане в Чатбота на потребителя се дава възможност да избере командата **/помощ** за да види как ефективно да комуникира с приложението.

  
Фигура 1. Начален екран - <http://localhost:12345/>. Опция за информация.

Първоначално Чатботът е настроен да отговаря на въпроси по специалност СОФТУЕРНИ ТЕХНОЛОГИИ СЪС СПЕЦИАЛИЗАЦИЯ СОФТУЕРНИ АРХИТЕКТУРИ И СРЕДСТВА – Срок на обучение 2 години – ІІ курс – зимен семестър.  
Програмата, за която потребителя може да задава своите въпроси се променя със следната последователност от заявки:

1. **Технологии и реализация**
2. **Back-end**

Python е основният език за програмиране на бекенд частта на проекта. Той е широко използван заради своята простота, четимост и голяма екосистема от библиотеки, които ускоряват разработката на различни видове приложения. За този проект Python се използва за събиране на данни, обработка на запитвания от потребителите и управление на логиката на чатбота.

**Използвани библиотеки:**

* **requests**: Тази библиотека се използва за изпращане на HTTP заявки. Тя е лесна за използване и осигурява надеждно извличане на съдържание от уебсайтове, което е необходимо за процеса на скрейпинг. В проекта тя се използва за получаване на HTML съдържание от уебсайта на Пловдивския университет.
* **BeautifulSoup (от библиотеката bs4)**: BeautifulSoup е библиотека за парсване на HTML и XML документи. Тя се използва за анализ на структурата на уеб страниците и извличане на данни като текст, линкове, изображения и други елементи. В проекта тя се използва за извличане на информация от HTML страниците на уебсайта на университета, включително специалности, програми и дисциплини.
* **Flask**: Flask е микрофреймуърк за уеб приложения на Python. Той е лек и лесен за използване, което го прави отличен избор за малки и средни проекти, като този чатбот. Flask управлява сървъра, обработва HTTP заявки и дефинира API точки за комуникация с фронтенд частта на приложението. Той също така осигурява възможности за маршрутизиране на заявките, обработка на POST и GET заявки и генериране на JSON отговори, които се връщат на потребителя.
* **socketserver и http.server**: Тези библиотеки са използвани за създаване на HTTP Proxy сървър, който интегрира чатбота в уебсайта на университета. Чрез тях е възможно модифицирането на съдържанието на сайта в реално време, като се добавя JavaScript, който позволява на потребителите да стартират чатбота директно през уебсайта на университета.

**Предимства на Python:**

* **Бърза разработка**: Python е известен със своето лесно усвояване и краткост на кода, което позволява бърза разработка на приложения.
* **Широка екосистема от библиотеки**: Множество библиотеки, като requests, BeautifulSoup, и Flask, които са широко използвани и добре поддържани, значително ускоряват разработката.
* **Добра поддръжка за уеб скрейпинг**: Python предоставя няколко библиотеки за уеб скрейпинг, които улесняват извличането на данни от уебсайтове, както е направено в този проект за събиране на информация за специалности и програми от сайта на Пловдивския университет.

* 1. **Front-end**

Angular предлага множество мощни функционалности за създаване на динамични уеб приложения. Сред тях са:

* **Компоненти**: Angular приложенията се базират на компоненти, които представляват отделни части от потребителския интерфейс. Всяка компонента има своето собствено съдържание и логика, като в проекта на чатбота, компонентите управляват входа от потребителя и показването на съобщения.
* **Директиви**: Angular предоставя директни инструменти за манипулиране на DOM елементи в HTML, като например \*ngIf и \*ngFor, които се използват за условно показване на елементи и итерация през списъци.
* **Двустранно свързване на данни**: Системата за двустранно свързване на данни в Angular позволява автоматично актуализиране на интерфейса, когато се променят данни в модела, и обратно.
* **Услуги (Services)**: Angular използва услуги за да инкапсулира логиката, която е споделена между различни компоненти. В този проект например се използва услуга ChatbotApiClientService, която се грижи за извършването на HTTP заявки към бекенд сървъра и връщането на отговори.

**Използвани библиотеки и технологии**

* **HttpClient**: Това е основната библиотека за изпращане на HTTP заявки в Angular приложенията. В проекта тя се използва за комуникация с API-то на бекенда, като се изпращат въпроси от потребителя и се получават отговори от чатбота. Чрез HttpClient се правят POST заявки към сървъра, които съдържат въпросите на потребителя.
* **RxJS**: RxJS (Reactive Extensions for JavaScript) е библиотека, която Angular използва за асинхронно програмиране. В този проект се използва за работа с наблюдаеми обекти (Observable), които осигуряват обработка на асинхронни събития като отговори от сървъра.
* **FormsModule**: Това е Angular модул, който се използва за работа с форми и свързване на входни данни с модели. В проекта, чрез ngModel, се свързва потребителският вход с данни в компонентите, което позволява динамично актуализиране на стойностите.
* **TypeScript**: Angular е построен върху TypeScript, разширение на JavaScript, което добавя типова система и поддръжка за обектно-ориентирано програмиране. TypeScript осигурява по-добра поддръжка на инструменти за разработка, проверка на типове по време на компилация и по-добро структуриране на приложенията.

**Работа с компоненти**

* **Примерен код за компонента AppComponent:**
* **Основна структура на компонента**:
* Компонентът използва ChatbotApiClientService за извършване на HTTP заявки към бекенд сървъра.
* В компонентата се запазва историята на чат съобщенията (chatHistoryStack), която се показва на екрана. Потребителят може да въвежда въпроси, които се изпращат към сървъра и се добавят към историята на чат съобщенията.
* Използва се директивата ngFor, за да се показват всички съобщения в историята на чата.

**Двустранно свързване на данни**:

* Потребителският вход е свързан с променливата userChatInput чрез ngModel, което позволява лесно проследяване и актуализиране на стойността на текстовото поле за въвеждане.

**Методи за работа с потребителски вход и API комуникация**:

* sendMessage() се извиква, когато потребителят изпрати съобщение. Той изпраща въпроса към бекенд сървъра чрез метода getPrograms() на услугата ChatbotApiClientService и добавя отговорите към историята на чата.
* Методът onKeydown() хваща натисканията на клавиша Enter, за да изпрати съобщение без да преминава към нов ред в текстовото поле.

**Скролиране на чата**:

* Методът scrollToBottom() се използва за автоматично скролиране до последното съобщение в чата.

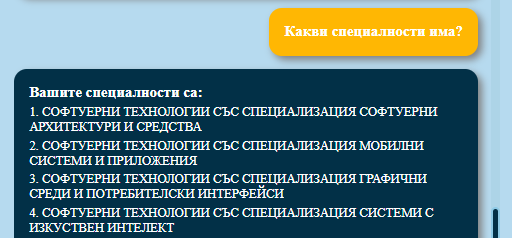
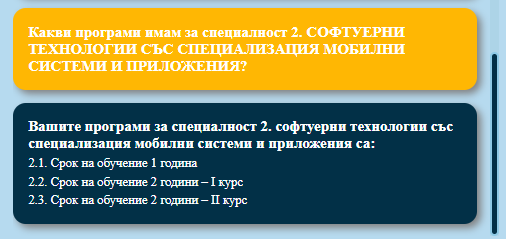
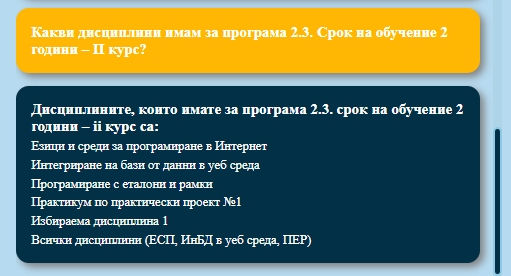
**Използване на Angular Routing**

* В проекта може да се използва Angular Routing, за да се добави навигация между различни изгледи, но в този случай, проектът е фокусиран върху един основен компонент (AppComponent), който управлява всичките потребителски взаимодействия.

**Предимства на Angular в този проект**

* **Мощна интеграция с HTTP API**: Angular предоставя лесен начин за извършване на HTTP заявки и интеграция с RESTful API-та, което е изключително полезно в този проект, тъй като чатботът комуникира със сървъра чрез POST заявки.
* **Компонентна структура**: Чрез използването на компоненти, Angular предлага чист и структуриращ начин за изграждане на интерфейса, като всеки компонент може да отговаря за своята част от логиката и изгледа.
* **Поддръжка на асинхронни операции**: RxJS и HttpClient позволяват ефективно управление на асинхронни задачи, като например обработка на отговори от сървъра, което е основополагащо за приложението на чатбота, което работи с API.

1. **Демонстрация**

* **Какви специалности има?** – връща списък с всички магистърски програми на ФМИ.  
  
* **Какви програми имам за специалност 2. СОФТУЕРНИ ТЕХНОЛОГИИ СЪС СПЕЦИАЛИЗАЦИЯ МОБИЛНИ СИСТЕМИ И ПРИЛОЖЕНИЯ?** – показва наличните   
  курсове по избраната специалност. 
* **Какви дисциплини имам за програма 2.3. Срок на обучение 2 години – ІІ курс?** – Връща дисциплините от текущия семестър на избрания курс  
  
* **Въпроси по избраната програма**

