****

**ИКОНОМИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА  
ФАКУЛТЕТ „ИНФОРМАТИКА“**

**Курсова работа**

**на тема**

**„Система за CRUD операции за уеб приложение за споделяне на пътувания“**

***по дисциплината Клиентско уеб програмиране***

**Изготвил: Проверил:**

Ния Димитрова Гл. ас. д-р Б. Банков

Фак. № 124157 Х. ас. Н. Цанков

Спец. МУТ, гр. 37

ВАРНА 2025

Съдържание

[1. Въведение 3](#_Toc199325502)

[2. Структура на проекта 4](#_Toc199325503)

[3. Преглед на използваните функции за извършване на CRUD операции 6](#_Toc199325504)

[3.1 Метод submitTrip(e) 6](#_Toc199325505)

[3.2 Метод deleteUser(e) 7](#_Toc199325506)

[3.3 Метод showDashboardView 8](#_Toc199325507)

[3.4 Метод showDetailsView(ctx) 8](#_Toc199325508)

[3.5 Метод deleteTrip(e) 9](#_Toc199325509)

[3.6 Метод toggleGoing(e) 10](#_Toc199325510)

[3.7 Метод updateTrip(e) 11](#_Toc199325511)

[3.8 Метод signup(e) 11](#_Toc199325512)

[3.8 Метод signin(e) 12](#_Toc199325513)

[3.9 Метод showMyDashboardView(ctx) 13](#_Toc199325514)

[3.10 Метод showSearchView(ctx) 13](#_Toc199325515)

[4. Заключение 14](#_Toc199325516)

# Въведение

Целта на проекта е да се създаде система за CRUD операции за уеб приложението “You’ll come tool”. Уеб приложението представлява платформа за споделяне на информация и идеи за пътувания между приятели и познати. Функционалностите на уеб страницата включват създаване на потребителски профили, админ панел, през който могат да се изтриват потребители и да се манипулират вече създадени пътувания, качване на пълна информация за пътувания – дати, данни за полети, място на преспиване, транспорт и т.н., редактиране и изтриване на пътувания, списък с идващи, страница с всички пътувания, страница с пътувания, създадени от регистриралия се потребител, опция за търсене на конкретно пътуване по име. Уеб приложението използва SQLite база данни и фреймуорка Flask, за да обработва клиентските заявки и да съхранява данни.

Уеб приложението е SPA – Single Page Application, съответно зарежда различно съдържание в една и съща страница, която в конкретния случай е index.html. В клиентската част се използват библиотеката Page.js за по-лесно рутиране между страниците и Lit.html за по-лесно и бързо зареждане на съдържанието.

Системата за CRUD операции е слоят между потребителския интерфейс и сървърната част. Тя се състои от отделни JavaScript файлове, които са отговорни за зареждането на конкретна страница, зареждането на динамични данни от базата данни и изпращане на нови данни към базата. Тя е нужна, за да е възможна работата на приложението.

# Структура на проекта

Структурата е както следва:

1. Папка .vscode – JSON конфигурационни файлове за стартиране на приложението
2. Папка back-end
   1. app.db – базата данни
   2. app.py – тук се съдържат всички функции, които обработват клиентските заявки, проверяват валидността на данните, модифицират базата данни
   3. storage.py – чрез този файл се създава базата данни и нейната структура, чрез изпълнение на отделни SQL команди
   4. Папка \_\_pycache\_\_ - кеширани данни
3. Папка front-end
   1. Папка images - изображения
   2. Папка node\_modules – изтеглени модули
   3. Папка src
      1. Папка views – тук се съдържат всички файлове, които променят съдържанието на страницата и изпращат HTTP заявки
         1. addTrip.js – код на страницата за добавяне на ново пътуване
         2. admin.js – код на админ панела
         3. congrats.js – код на страницата, след присъединяване към пътуване
         4. dashboard.js – код на страницата с всички пътувания
         5. details.js – код на страница с детайли за конкретно пътуване
         6. edit.js – код на страница за редактиране на конкретно пътуване
         7. home.js – код на началната страница
         8. login.js – код на страницата за вписване на потребител
         9. myTrips.js – код на страницата на пътуванията на текущо вписания потребител
         10. register.js – код на страницата за регистриране на нов потребител
         11. search.js – код за извеждане на всички пътувания, чиито имена отговарят на заявката за търсене
      2. app.js – тук се рутират всички изгледи и се управляват бутона за излизане от профила и полето за търсене
      3. lib.js – тук се импортират библиотеки и се експортират с по-кратки имена нужните функции
      4. util.js – функции за управление на потребителски данни и управление на лентата за навигация
   4. Папка styles – тук се съдържат всички CSS файлове за отделните страници
   5. package-lock.json
   6. package.json
4. index.html – главната страница, където се зарежда съдържанието

# Преглед на използваните функции за извършване на CRUD операции

## Метод submitTrip(e)

Фиг. 1 – Записване на данните от формулярите в обект и проверка на полетата

Фиг. 2 – Изпращане на данните към сървъра

Методът submitTrip се извиква при натискане на submit бутона на формуляра на страницата за добавяне на ново пътуване. Функцията приема един аргумент – събитието на натискането. Първо функцията взима и запазва в константи данните от четирите формуляра – basicsForm, flightForm, transportForm, accomodationForm. Данните се взимат от FormData обекти, които са създадени чрез формулярите. След тяхното запазване и проверка на валидността на датите, данните се запазват в нов обект – tripObj. Прави се проверка дали всички полета са попълнени. Създава се нов FormData обект, в който се прехвърлят данните от полетата и каченото изображение за корица на пътуването. Данните се изпращат към сървърната част чрез POST заявка и потребителят бива прехвърлен на началната страница.

## A screen shot of a computer code AI-generated content may be incorrect.Метод deleteUser(e)

Фиг. 3 – Методът deleteUser

Методът deleteUser служи за изтриване на потребител чрез админ панела. Първо взима userId от HTML-а на страницата. При зареждане на админ панела се изреждат всички потребители един под друг с отделни бутони за изтриване. В dataset атрибутите на бутоните се съдържа ID на конкретния потребител, освен ако не е админ. ID-то на избрания потребител се добавя към линк и се изпраща DELETE заявка към сървъра. В случай, че изтриването е успешно, страницата се презарежда.

A screenshot of a web page

AI-generated content may be incorrect.

Фиг. 4 – Админ панела

## A screen shot of a computer program AI-generated content may be incorrect.Метод showDashboardView

Фиг. 5 – Методът showDashboardView

Методът showDashboardView изпраща GET заявка към сървъра. След това подава получените данни на функцията, която създава dashboard шаблона, чрез библиотеката lit-html. След това тази функция се подава на функцията render, която закача съдържанието в главната страница. В случай на грешка се извежда съобщението на грешката.

## Метод showDetailsView(ctx)

Методът showDetailsView(ctx) приема един аргумент – контекст, чрез който взима tripId от адреса на страницата. Изпраща GET заявка към сървъра, за да получи данни за избраното пътуване. В случай, че в local storage има запазени данни за потребител, те биват запазени в променливи. Извършва смятане на пълната цена на пътуването с част от получените данни. След това изпраща втора GET заявка, за да получи данни за списъка от отиващи на пътуването потребители. Проверява дали текущо влезлият потребител е в списъка. Рендира секцията в главната страница чрез шаблон, на който се подава списък с нужните аргументи. Накрая се добавят event listeners на бутоните за изтриване на пътуване, за присъединяване към списъка на отиващите и бутонът за копиране на линк.

Фиг. 6 – Метод showDetailsView

## Метод deleteTrip(e)

Методът deleteTrip използва глобалната променлива tripIdg (tripIdGlobal) и я добавя към URL-а, на който изпраща DELETE заявка. В случай, че е успешна, потребителят се пренасочва на актуализираната страница с всички пътувания.

## Метод toggleGoing(e)

A screen shot of a computer program

AI-generated content may be incorrect.Методът toggleGoing приема един аргумент – събитието, чрез което е извикан. Първо проверява дали бутонът за присъединяване към списъка на идващите има клас „going“. В зависимост дали има, изпраща DELETE или POST заявка към съответния URL. Ако потребителят напуска списъка на отиващите, бива пренасочен към страницата с всички пътувания. В случай, че се присъединява, бива пренасочен към поздравителна страница. И в двата случая бутонът бива променен. В session storage се запазва tripId, за да може след пренасочване към поздравителната страница да се разбере към кое пътуване се е присъединил потребителят.

Фиг. 7 – Метод toggleGoing

## Метод updateTrip(e)

A screen shot of a computer program

AI-generated content may be incorrect.Методът updateTrip извършва същите операции като метода submitTrip, с разликата, че данните във формулярите изначално се зареждат чрез GET заявка. След като потребителят редактира пътуването, всички данни се изпращат към сървъра с PUT заявка, за да може само променените данни да се актуализират.

Фиг. 8 – Кодът за проверка на данните и за изпращане на PUT заявка

## 3.8 Метод signup(e)

Методът signup приема като аргумент само събитието, чрез което е извикан. В случай, че в local storage няма данни за user, се създава обект user с полета firstName, lastName, email и password, като техните стойности са undefined. Ако има потребителски данни в local storage, то тогава те се запазват в променливата user. Взимат се стойностите на полетата за парола и за повторно изписване на парола. Проверява се дали полето за паролата не е оставено празно и дали двете пароли съвпадат. Проверява се дали са приети условията за ползване и поверителност. В обекта се записват стойностите на полетата firstname, lastname, email и password. Изпраща се POST заявка с обекта към сървъра. В случай, че няма грешка, нужните данни за новия потребител се записват в local storage, лентата за навигация се обновява и потребителят се препраща на началната A screen shot of a computer code

AI-generated content may be incorrect.страница.

Фиг. 9 – Изпращане на POST заявката и записване на получените данни в local storage

## Метод signin(e)

Методът signin приема един аргумент – събитието, чрез което е извикан. Първо записва в променлива референция към полето за грешка и взима имейла и паролата от input полетата. Проверява дали не са празни. В случай, че не са, ги записва в обект user и изпраща обекта към сървъра с POST заявка. При успешно вписване, сървърът връща имейл, userId и булева променлива isAdmin. Проверява се дали текущо вписаният потребител е админ. Ако е, се прехвърля към админ панела и в local storage се запазват имейлът, userId и променливата isAdmin със стойност true. Ако потребителят не е админ, в local storage се запазват само имейл и userId, лентата за навигация се обновява и потребителят се прехвърля на началната страница. В случай на грешка, съобщението за грешка се показва и цветът му се сменя на червен.

## Метод showMyDashboardView(ctx)

Методът showMyDashboardView взима потребителските данни, записани в local storage и записва userId в променлива. След това изпраща GET заявка към сървъра, като включва и userId в URL-а. Накрая подава получените данни на функцията, която зарежда шаблона на страницата и след това тази функция се подава на функцията render, която зарежда съдържанието в главната страница.

## Метод showSearchView(ctx)

A close-up of a text

AI-generated content may be incorrect.Методът showSearchView приема един аргумент – контекст, чрез който взима заявката, по която трябва да търси, от URL-а. Изпраща GET заявка към сървъра. Презарежда dashboard страницата само с пътуванията, които отговарят на заявката.

Фиг. 0 – Данни, които връща сървъра при успешно изпращане на заявката

# Заключение

Уеб приложението има пълен набор от функции, за да извършва нормалната си дейност. Могат да се изпращат всички видове заявки към сървъра. От сървъра може и да се получават данни, които биват правилно обработвани и визуализирани. Ефектът на заявките е дълготраен, понеже данните се пазят в локална база данни.