

Документација за лабораториска вежба II

Слики и објаснувања за кодот

```
app.config['MAIL_SERVER'] = 'smtp.sendgrid.net'
app.config['MAIL_PORT'] = 587
app.config['MAIL_USERNAME'] = 'apikey'
app.config['MAIL_PASSWORD'] = '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'
app.config['MAIL_USE_TLS'] = True
app.config['MAIL_USE_SSL'] = False
app.config['MAIL_DEFAULT_SENDER'] = 'tamarageorgieva2001@gmail.com'
app.secret_key = os.urandom(24)
```

Во првиот дел од кодот ги конфигурирам поставките за е-пошта. Се поставуваат подесувањата за SMTP серверот на sendgrid како што се порта, информации за автентикација (корисничко име и лозинка), како и се наведува користење на TLS и се дефинира стандардна е-пошта од која ќе бидат испратени е-маилите.

```
def send_email(email, subject, content):
    message = Mail(
        from_email=app.config['MAIL_DEFAULT_SENDER'],
        to_emails=email,
        subject=subject,
        html_content=content
    )

    try:
        sg = SendGridAPIClient(app.config['MAIL_PASSWORD'])
        response = sg.send(message)
        if response.status_code == 202:
            return True
        else:
            return False

    except Exception as e:
        traceback.print_exc()
        return False
```

Оваа функција претставува делот за испраќање на е-пораки преку SendGridAPI во Flask апликацијата. Функцијата ги користи конфигурациските поставки за е-пошта кои ги поставив во предходната слика.

```
def create_users_file():
    desktop_path = os.path.join(os.path.expanduser('~'), 'Desktop')
    file_path = os.path.join(desktop_path, 'users.txt')
    if not os.path.exists(file_path):
        with open(file_path, 'a+'):
            pass
    return file_path
```

Оваа функција, `create_users_file()`, служи за креирање на текстуална датотека (`users.txt`) на Desktop, доколку не постои таква датотека, потоа ја отвора во режим за читње и пишување, доколку веќе постои, само ја отвора во веќеспоменатиот режим.

```
def hash_password(password):
    return bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt()).decode('utf-8')
```

Оваа функција, `hash_password(password)`, е дизајнирана да ја хешира корисничка лозинка со корисрење на `bcrypt` алгоритмот. Овој алгоритам работи на принципот на првично претворање на пасвордот во битови, потоа генерира и додава “сол” на пасвордот за поголема безбедност и на крајот го враќа пасвордот како стринг од хешираната верзија на пасвордот.

```
def generate_verification_code(email):
    return secrets.randbelow(1000000)
```

Оваа функција генерира шестцифрен број кој се користи како верификациски код за дадената е-пошта.

```

def register(username, password, email):
    file_path = create_users_file()

    with open(file_path, 'r') as file:
        lines = file.readlines()
        for line in lines:
            stored_info = line.strip().split(':')
            if len(stored_info) >= 4:
                stored_username, _, stored_email, _ = stored_info[:4]
                if stored_username.lower() == username.lower():
                    return "Registration failed: Username already exists. Please choose another one."
                if stored_email.lower() == email.lower():
                    return "Registration failed: Email already exists. Please enter another email."

    if not re.match(pattern: r'^[\w.-]+@[\w.-]+\.[a-zA-Z]{2,}$', email):
        return "Registration failed: Invalid email address. Please enter a valid email address."

    hashed_password = hash_password(password)

    if not all([
        re.search(pattern: r'[A-Z]', password),
        re.search(pattern: r'[!@#$%^&*()_+{}[\]|:;<.>./~]', password),
        re.search(pattern: r'[0-9]', password)
    ]):
        return "Password must contain at least one uppercase letter, one symbol, and one digit. Please choose another password."

    verification_code = generate_verification_code(email)
    if not send_email(email, subject: 'Your Verification Code', content: f'Your verification code is: {verification_code}'):
        return "Failed to send verification code"

    with open(file_path, 'a') as file:
        file.write(f'{username.lower()}:{hashed_password}:{email}:{verification_code}\n')

    return "User registered successfully!"

```

Оваа функција register има за цел да регистрира нови корисници. Таа проверува во текстуална датотека (users.txt) дали корисничкото име или е-поштата веќе постојат, го валидира форматот на е-поштата и лозинката, генерира верификациски код и праќа е-пошта за потврда на корисникот. Ако сè е успешно, регистрира нов корисник и го зачувува во датотеката. Ако настане некоја грешка, враќа соодветна порака за неуспешна регистрација.

```
def login(username, password, entered_code):
    file_path = create_users_file()

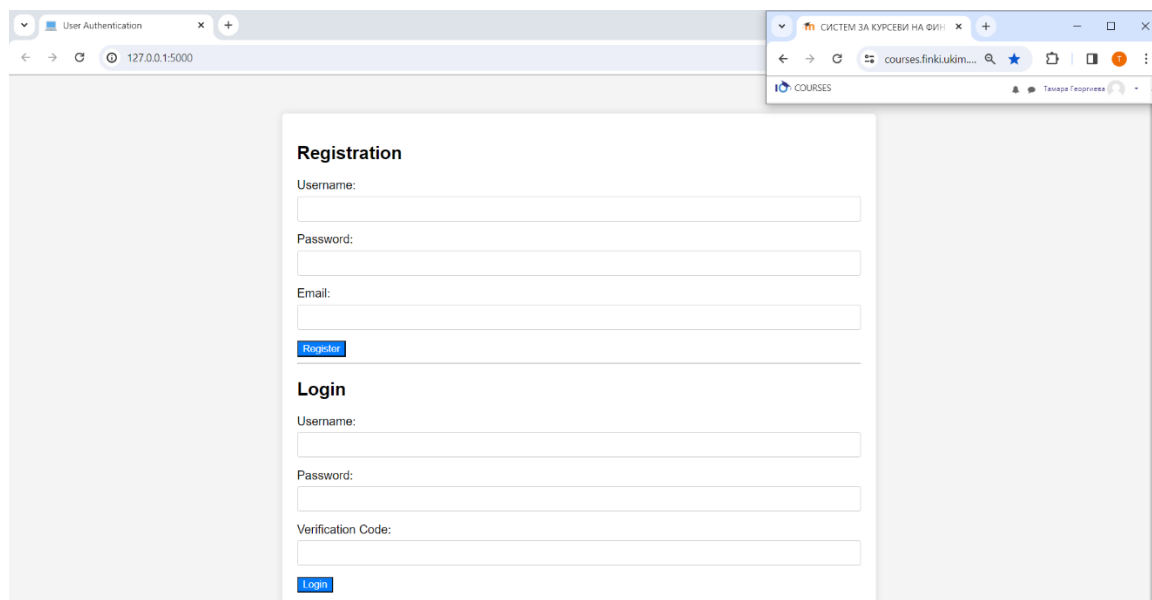
    with open(file_path, 'r') as file:
        for line in file:
            stored_username, stored_password, stored_email, stored_code = line.strip().split(':')
            if stored_username.lower() == username.lower():
                if verify_password(password, stored_password):
                    if entered_code == stored_code:
                        return "Login successful!"
                    else:
                        return "Invalid verification code."
                else:
                    return "Invalid username or password."

    return "Invalid username or password."
```

Оваа функција прво ги чита податоците од текстуалната датотека (users.txt) каде што се чуваат информации за корисниците. За секој ред од датотеката, ги споредува внесеното корисничко име со зачуваното во датотеката. Ако корисничкото име е пронајдено, функцијата проверува дали внесената лозинка се совпаѓа со зачуваната лозинка за тоа корисничко име. Ако лозинката е валидна, проверува дали внесениот верификациски код се совпаѓа со зачуваниот за истиот корисник. Во зависност од внесеното, враќа соодветна порака за успешна или неуспешна најава.

Слики од изгледот и работата на веб страната

Воведна страна



The screenshot displays two web browser windows. The left window, titled 'User Authentication', shows a local server address '127.0.0.1:5000'. The right window, titled 'СИСТЕМ ЗА КУРСОВИ НА ОБИД', shows the URL 'courses.finki.ukim...'. Both windows display a web page with two main sections: 'Registration' and 'Login'. The 'Registration' section includes input fields for 'Username:', 'Password:', and 'Email:', followed by a blue 'Register' button. The 'Login' section includes input fields for 'Username:', 'Password:', and 'Verification Code:', followed by a blue 'Login' button.

Тамара Георгиева 213207

Изглед на пополнет дел за регистрација

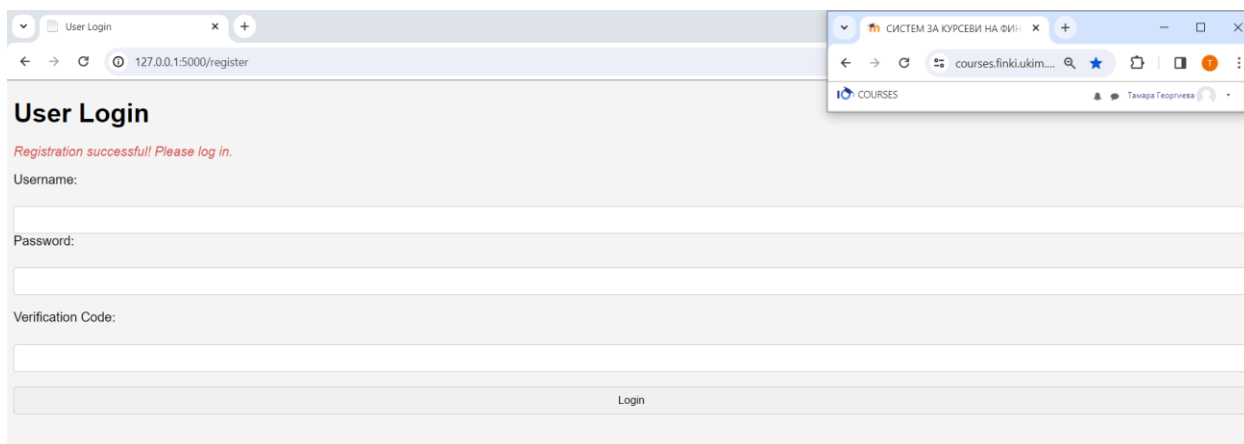


The screenshot shows a web browser window with the title "User Authentication" and the URL "127.0.0.1:5000". The page displays a "Registration" form with the following fields:

- Username:** tamarageorgievag
- Password:** (masked with dots)
- Email:** tamarageorgievag@gmail.com

A blue "Register" button is located at the bottom of the form.

Откако корисникот успешно ќе се регистрира, веднаш добива можност да се најави

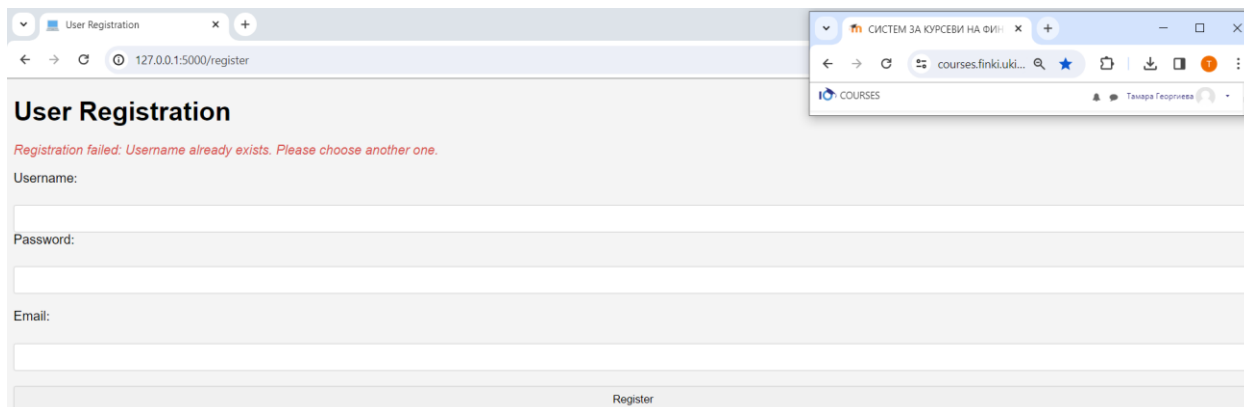


The screenshot shows a web browser window with the title "User Login" and the URL "127.0.0.1:5000/register". The page displays a "User Login" form with the following fields:

- Username:** (empty)
- Password:** (empty)
- Verification Code:** (empty)

A "Login" button is located at the bottom of the form. A red message above the form reads: "Registration successful! Please log in."

Ако корисникот не се регистрира точно, веднаш ја добива опција повторно да се обиде да се регистрира



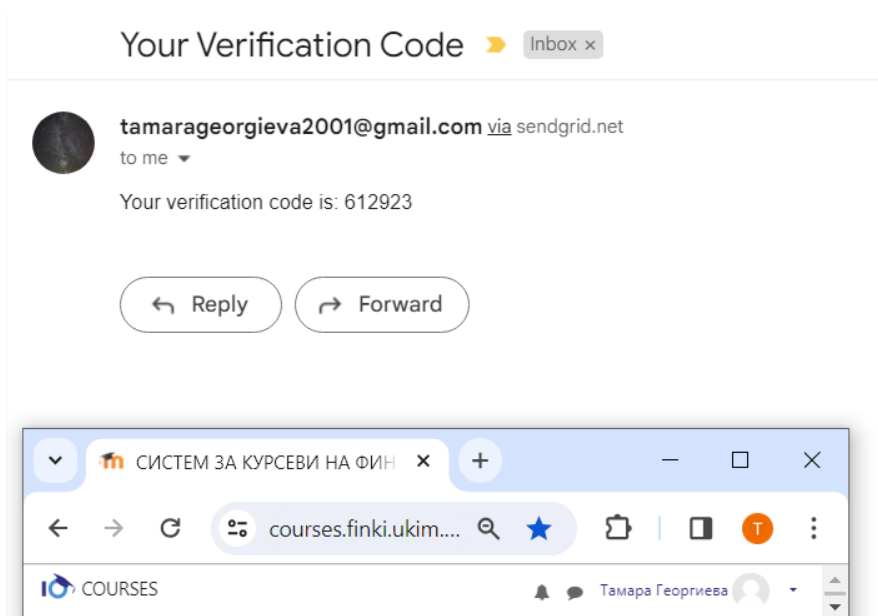
The screenshot shows a web browser window with the title "User Registration" and the URL "127.0.0.1:5000/register". The page displays a "User Registration" form with the following fields:

- Username:** (empty)
- Password:** (empty)
- Email:** (empty)

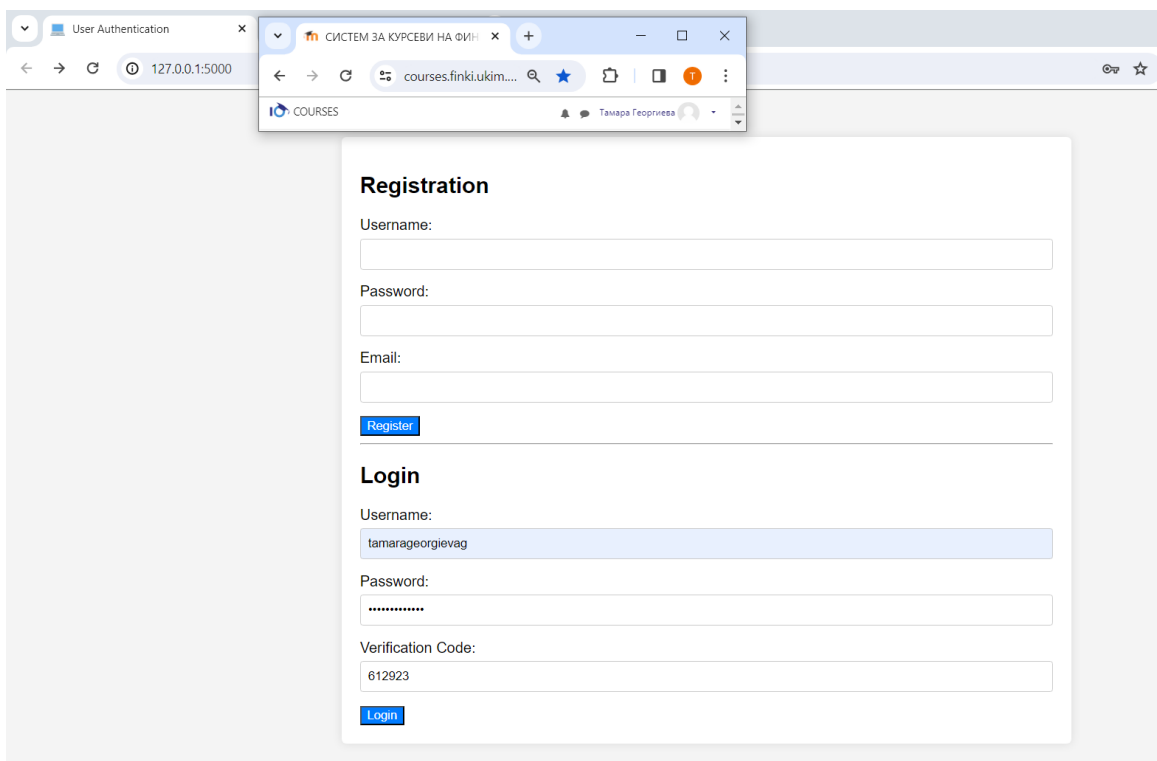
A "Register" button is located at the bottom of the form. A red message above the form reads: "Registration failed: Username already exists. Please choose another one."

Тамара Георгиева 213207

По успешна регистрација, корисникот добива верификациски код на емаил.



Откако корисникот ќе го добие кодот за регистрација, тој/таа може да се логира во системот со своето корисничко име, пасворд и верификацискиот код.



Тамара Георгиева 213207

Ако корисникот направи грешка при логирањето, системот ќе му каже дека има грешка и ќе му се даде можност на повторно да се обиде да се логира.

User Login

Invalid username or password.

Username:

Password:

Verification Code:

Login

Ако корисникот успешно се логира, добива порака за добредојде со нивното корисничко име.

Welcome, tamarageorgievag!

COURSES