TRƯỜNG ĐẠI HỌC VĂN LANG KHOA CÔNG NGHỆ THÔNG TIN





BÁO CÁO ĐÔ ÁN MÔN HỌC LẬP TRÌNH PYTHON NÂNG CAO

Chủ đề:

WEB APPLICATIONS QUẢN LÍ THƯ VIỆN

SVTH: Châu Gia Kiệt - 2274802010449

Võ Đình Ngọc Bình – 2274802010066

LHP: 241_71ITSE31003_0203

GVHD: ThS Trần Kim Mỹ Vân

TP. Hồ Chí Minh – năm 2024

Mục Lục

LỜI NÓI ĐÂU	1
CHƯƠNG 1: TỔNG QUAN	2
1.1 Mục tiêu của dự án	2
1.2 Giới thiệu về Flask	2
1.3 Lý do chọn Flask cho dự án	2
1.4 Giới thiệu về Docker	2
1.5 Úng dụng Docker vào FitLab	3
1.6 Công nghệ và công cụ sử dụng	3
CHƯƠNG 2: KẾT LUẬN	4
2.1 Kết quả đạt được	4
2.2 Khó khăn gặp phải và cách khắc phục	4
2.3 Bài học kinh nghiệm	4
2.4 Hướng phát triển trong tương lai	4
CHƯƠNG 3: DEMO	6
3.1 Giao diện chương trình	6
3.1.1 Trang chủ	6
3.1.2 Trang đăng nhập	7
3.1.3 Trang đăng kí	7
3.1.4 Trang thuê sách	7
3.1.5 Trang sách đã thuê	8
3.1.6 Trang thêm sách	8
3.1.7 Trang sửa sách	9
3.2 Mã chương trình	9

DANH MỤC HÌNH ẢNH

Fình 1. Giao diện trang chủ	6
Tình 2. Giao diện trang chủ khi đăng nhập	6
Tình 3. Giao diện trang đăng nhập	7
Tình 4. Giao diện trang đăng kí	7
Fình 5. Giao diện trang thuê sách	7
Fình 6. Giao diện trang sách đã thuê	8
Fình 7. Giao diện trang thêm sách	8
Iình 8. Giao diện trang chỉnh sửa sách	9

LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin phát triển vượt bậc, việc xây dựng các ứng dụng web không chỉ mang lại những tiện ích thiết thực mà còn mở ra nhiều cơ hội học tập và ứng dụng các kỹ năng mới. Qua môn học này, em đã được tiếp cận với framework Flask – một công cụ phát triển ứng dụng web đơn giản nhưng rất hiệu quả. Bằng việc áp dụng những kiến thức lý thuyết vào thực tế, em đã từng bước phát triển một ứng dụng web có tính năng phù hợp với yêu cầu của bài tập cuối kỳ, đồng thời giúp em củng cố và mở rộng kiến thức trong lĩnh vực lập trình web.

Và đồng thời em cũng xin gửi lời cảm ơn chân thành đến **ThS Trần Kim Mỹ Vân** đã nhiệt tình hướng dẫn, giải đáp mọi thắc mắc và đưa ra những lời khuyên quý báu trong suốt quá trình thực hiện bài tập. Chính nhờ sự hỗ trợ và định hướng của cô mà em đã vượt qua được những khó khăn trong quá trình nghiên cứu và phát triển dự án này. Em mong rằng bài báo cáo sẽ là minh chứng cho những cố gắng và kết quả đạt được từ sự hướng dẫn tận tâm của cô, cũng như từ sự nỗ lực của bản thân trong suốt học kỳ vừa qua. Nếu còn sai sót gì trong quá trình làm bài mong quý thầy cô và các bạn lượng thứ bỏ qua.

Xin chân thành cảm ơn quý thầy/cô và các bạn.

CHƯƠNG 1: TỔNG QUAN

1.1 Mục tiêu của dự án

Mục tiêu chính của dự án này là phát triển một ứng dụng web RESTful sử dụng Flask - một framework phổ biến và nhẹ trong Python. Ứng dụng có chức năng quản lý các thao tác cơ bản như thêm, sửa, xóa và tìm kiếm thông tin một cách dễ dàng và hiệu quả. Dự án giúp sinh viên thực hành và áp dụng kiến thức về lập trình web và phát triển API, từ đó nâng cao kỹ năng lập trình và phát triển phần mềm.

1.2 Giới thiệu về Flask

Flask là một micro-framework dành cho Python, được thiết kế để tạo các ứng dụng web nhỏ, linh hoạt và dễ mở rộng. Flask có các đặc điểm nổi bật như:

- **Dễ học và dễ triển khai**: Với cấu trúc đơn giản, Flask giúp người mới bắt đầu có thể dễ dàng tiếp cận và xây dựng ứng dụng.
- **Tính linh hoạt cao**: Cho phép tùy chỉnh theo nhu cầu của người phát triển, không bị ràng buộc vào những chuẩn mực quá cứng nhắc.
- Hỗ trợ phát triển RESTful API: Flask hỗ trợ mạnh mẽ cho việc xây dựng các API RESTful, phù hợp với các dự án yêu cầu tính linh hoạt và hiệu suất cao.

1.3 Lý do chọn Flask cho dự án

Flask là lựa chọn tối ưu cho dự án vì các lý do sau:

- **Cộng đồng hỗ trợ mạnh mẽ**: Với nhiều tài liệu và hướng dẫn trực tuyến, Flask giúp sinh viên tự học và phát triển.
- Khả năng mở rộng: Flask có thể được tích hợp dễ dàng với các công nghệ như SQLAlchemy để quản lý cơ sở dữ liệu, phù hợp cho việc phát triển các ứng dụng từ nhỏ đến lớn.
- Hiệu suất cao trong các ứng dụng nhỏ và vừa: Đối với các dự án học thuật, Flask cung cấp hiệu năng đủ mạnh và không yêu cầu tài nguyên hệ thống lớn.

1.4 Giới thiệu về Docker

Docker là một nền tảng mã nguồn mở cho phép đóng gói và triển khai các ứng dụng trong các container. Container là một môi trường ảo hoá nhẹ, giúp cách ly các ứng dụng với hệ

thống và môi trường bên ngoài. Điều này giúp việc phát triển, kiểm thử và triển khai ứng dụng trở nên linh hoạt và dễ dàng hơn. Docker mang lại một số lợi ích nổi bật như:

- **Tính nhất quán**: Docker đảm bảo rằng ứng dụng chạy đồng nhất trên mọi môi trường, từ máy tính của nhà phát triển đến các máy chủ sản xuất. Điều này giúp loại bỏ các vấn đề liên quan đến cấu hình khác nhau giữa các hệ thống.
- Khả năng mở rộng và triển khai nhanh chóng: Docker cho phép tạo nhiều container nhanh chóng, phù hợp với nhu cầu mở rộng của ứng dụng khi có lưu lượng truy cập tăng cao.
- Quản lý phụ thuộc dễ dàng: Docker đóng gói toàn bộ các phụ thuộc của ứng dụng,
 giúp tránh xung đột khi chạy các ứng dụng khác nhau trên cùng một máy chủ.

1.5 Úng dụng Docker vào FitLab

Docker là công cụ mạnh mẽ cho phép đóng gói và triển khai các ứng dụng trong các container, đảm bảo rằng chúng hoạt động ổn định và nhất quán trên mọi môi trường. Việc ứng dụng Docker vào fitlab giúp các dự án có thể dễ dàng triển khai và quản lý trên nền tảng này. Container hóa với Docker mang lại các lợi ích như: cách ly môi trường, tránh xung đột cấu hình, và dễ dàng mở rộng quy mô khi cần. Khi được triển khai trên fitlab, Docker giúp tối ưu hóa tài nguyên, giảm thời gian cài đặt và cấu hình, đồng thời tạo điều kiện thuận lợi cho việc kiểm thử và triển khai liên tục, phù hợp với nhu cầu học tập và phát triển các ứng dụng tại Văn Lang University.

1.6 Công nghệ và công cụ sử dụng

Trong dự án này, các công nghệ và công cụ chính được sử dụng bao gồm:

- Python: Ngôn ngữ lập trình chính để xây dựng ứng dụng.
- Flask: Framework chính để tạo API RESTful.
- PostgreSQL: Cơ sở dữ liệu để lưu trữ thông tin và quản lý dữ liệu.
- Docker

CHƯƠNG 2: KẾT LUẬN

2.1 Kết quả đạt được

Qua quá trình phát triển và hoàn thiện ứng dụng web RESTful này, dự án đã đạt được các kết quả sau:

- Xây dựng thành công với các chức năng cơ bản như tạo mới, cập nhật, xóa và lấy dữ liêu.
- Úng dụng được triển khai với cấu trúc rõ ràng, dễ dàng mở rộng và bảo trì.

2.2 Khó khăn gặp phải và cách khắc phục

Trong quá trình thực hiện dự án, một số khó khăn đã phát sinh, bao gồm:

- Cấu hình và tích hợp cơ sở dữ liệu: Việc thiết lập và quản lý cơ sở dữ liệu ban đầu gặp khó khăn, nhưng nhờ nghiên cứu tài liệu và sự hướng dẫn của giảng viên, em đã nắm vững cách làm việc với cơ sở dữ liệu trong Flask.
- Việc deploy với docker, trùng port khiên docekr gặp lỗi.

2.3 Bài học kinh nghiệm

Dự án không chỉ giúp sinh viên củng cố kiến thức về Flask mà còn cung cấp kinh nghiệm thực tiễn quý báu về lập trình web và phát triển ứng dụng. Qua dự án này, sinh viên đã hiểu rõ hơn về:

- Kỹ năng kiểm thử và gỡ lỗi ứng dụng.
- Tư duy logic và cách tổ chức cấu trúc mã nguồn một cách rõ ràng và dễ bảo trì.

2.4 Hướng phát triển trong tương lai

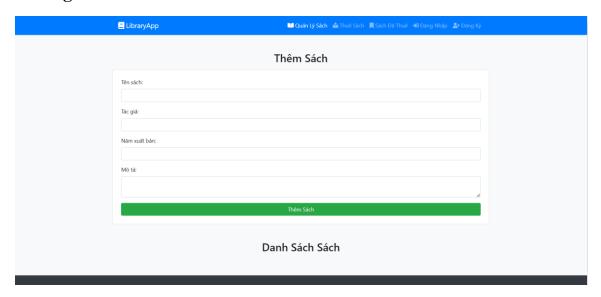
Trong tương lai, ứng dụng có thể được mở rộng với các tính năng và cải tiến sau:

- Bổ sung giao diện người dùng (UI): Xây dựng giao diện web để người dùng có thể tương tác trực quan với ứng dụng.
- Cải thiện bảo mật: Thêm các tính năng bảo mật cho ứng dụng, như xác thực người dùng, mã hóa dữ liệu.
- Triển khai lên nền tảng đám mây: Đưa ứng dụng lên các nền tảng như Heroku, AWS để nâng cao khả năng mở rộng và tiếp cận người dùng.

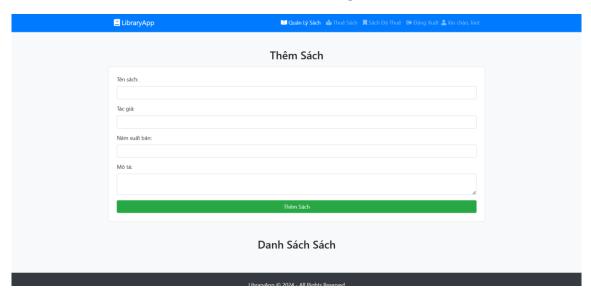
CHUONG 3: DEMO

3.1 Giao diện chương trình

3.1.1 Trang chủ

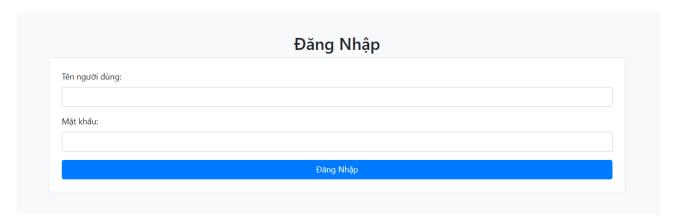


Hình 1. Giao diện trang chủ



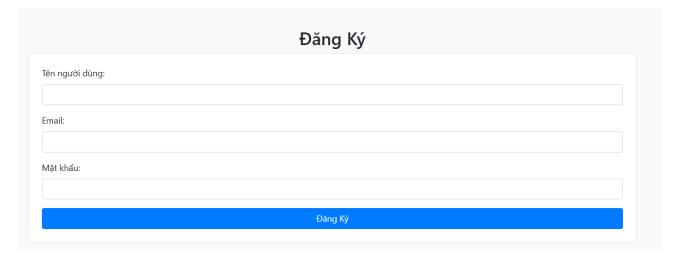
Hình 2. Giao diện trang chủ khi đăng nhập

3.1.2 Trang đăng nhập



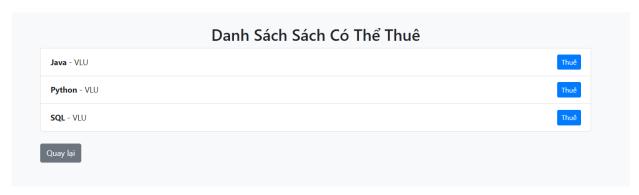
Hình 3. Giao diện trang đăng nhập

3.1.3 Trang đăng kí



Hình 4. Giao diện trang đăng kí

3.1.4 Trang thuê sách



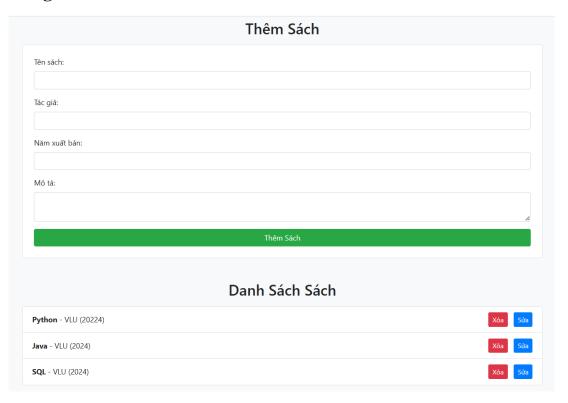
Hình 5. Giao diện trang thuê sách

3.1.5 Trang sách đã thuê



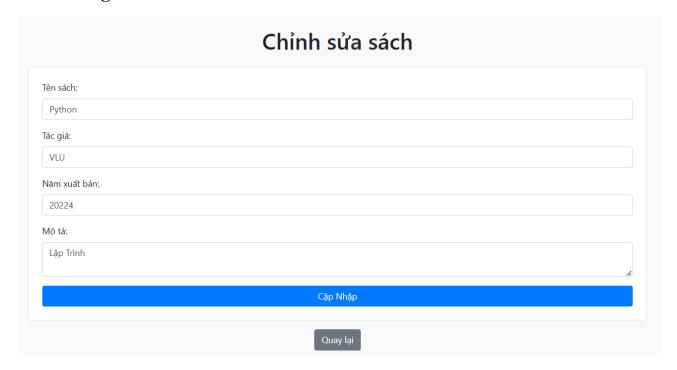
Hình 6. Giao diện trang sách đã thuê

3.1.6 Trang thêm sách



Hình 7. Giao diện trang thêm sách

3.1.7 Trang sửa sách



Hình 8. Giao diện trang chỉnh sửa sách

3.2 Mã chương trình

Link Github:

https://github.com/dustin314/Python_Programming/tree/main/Assignments/EXAM

---HÉT---

Bảng Phân Công Công Việc

Tên thành viên	Công việc
Châu Gia Kiệt	Viết code, viết file báo cáo
Võ Đình Ngọc Bình	Viết code

Mã code:

app.py

```
from flask import Flask, render_template, redirect, url_for, request, flash from models import db, Book, User, Rental
 from config import Config
from flask_login import LoginManager, login_user, logout_user, login_required, current_user from werkzeug.security import generate_password_hash, check_password_hash
app = Flask(__name__)
app.config.from_object(Config)
db.init_app(app)
migrate = Migrate(app, db)
login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'
@login_manager.user_loader
def load_user(user_id):
      return User.query.get(int(user_id))
@app.route('/')
      return render_template('index.html', books=books)
# Route thêm sách mới
@app.route('/add', methods=['POST'])
 @login_required
def add_book():
    title = request.form.get('title')
author = request.form.get('author')
published_year = request.form.get('published_year')
description = request.form.get('description')
      new_book = Book(title=title, author=author, published_year=published_year, description=description)
db.session.add(new_book)
      db.session.commit()
       return redirect(url_for('index'))
# Route xóa sách
@app.route('/delete/<int:id>', methods=['POST'])
@login_required
 def delete_book(id):
      book = Book.query.get(id)
      if book:
            db.session.delete(book)
            db.session.commit()
      return redirect(url_for('index'))
```

```
. .
    @app.route('/edit/<int:id>', methods=['GET'])
    @login_required
    def edit_book(id):
        book = Book.query.get(id)
        return render_template('edit.html', book=book)
   @app.route('/update/<int:id>', methods=['POST'])
    @login_required
   def update_book(id):
        book = Book.query.get(id)
        if book:
           book.title = request.form.get('title')
            book.author = request.form.get('author')
            book.published_year = request.form.get('published_year')
           book.description = request.form.get('description')
            db.session.commit()
        return redirect(url_for('index'))
   # Route đăng ký
@app.route('/register', methods=['GET', 'POST'])
   def register():
       if request.method == 'POST':
           username = request.form['username']
            email = request.form['email']
           password = request.form['password']
            # Kiểm tra người dùng đã tồn tại
            if User.query.filter_by(username=username).first() or User.query.filter_by(email=email).first():
                flash('Tên người dùng hoặc email đã tồn tại.')
                return redirect(url_for('register'))
            # Tạo người dùng mới
            new_user = User(username=username, email=email, password=generate_password_hash(password))
            db.session.add(new_user)
            db.session.commit()
            flash('Đăng ký thành công. Hãy đăng nhập.')
            return redirect(url_for('login'))
        return render_template('register.html')
```

```
@app.route('/login', methods=['GET', 'POST'])
   def login():
        if request.method == 'POST':
           username = request.form['username']
            password = request.form['password']
            user = User.query.filter_by(username=username).first()
            if user and check_password_hash(user.password, password):
                login_user(user)
                flash('Đăng nhập thành công.')
                return redirect(url_for('index'))
                flash('Tên người dùng hoặc mật khẩu không chính xác.')
        return render_template('login.html')
   @app.route('/logout')
   @login_required
   def logout():
        logout_user()
        flash('Bạn đã đăng xuất.')
        return redirect(url_for('login'))
   @app.route('/rent_books')
   @login_required
   def rent_books():
        available_books = Book.query.outerjoin(Rental).filter(
            (Rental.id == None) | (Rental.return_date != None)
        ).all()
        return render_template('rent_books.html', books=available_books)
   # Route để thuê sách
   @app.route('/rent/<int:book_id>', methods=['POST'])
   @login_required
   def rent(book_id):
       book = Book.query.get(book_id)
        if not book:
            flash('Book not found.')
            return redirect(url_for('rent_books'))
        rental = Rental(user_id=current_user.id, book_id=book.id)
        db.session.add(rental)
        db.session.commit()
        flash('You have successfully rented the book.')
        return redirect(url_for('my_rentals'))
```

```
@app.route('/my_rentals')
3 @login_required
4 def my_rentals():
       rentals = Rental.query.filter_by(user_id=current_user.id, return_date=None).all()
       return render_template('my_rentals.html', rentals=rentals)
10 @app.route('/return/<int:rental_id>', methods=['POST'])
11 @login_required
12 def return_book(rental_id):
        rental = Rental.query.get(rental_id)
        if rental and rental.user_id == current_user.id and rental.return_date is None:
           rental.return_date = datetime.utcnow()
           db.session.commit()
           flash('Trả sách thành công!')
            flash('Có lỗi xảy ra khi trả sách.')
       return redirect(url_for('my_rentals'))
23 if __name__ == "__main__":
       app.run(debug=True)
```

Config.py

```
import os

BASE_DIR = os.path.abspath(os.path.dirname(__file__))

class Config:
    SECRET_KEY = '73cdcb13b533e923f8e32a7586971c3f'
    SQLALCHEMY_DATABASE_URI = 'postgresql://postgres:123456@localhost:5432/dbbook'
    SQLALCHEMY_TRACK_MODIFICATIONS = False
```

create_user.py

```
1 from app import app, db
2 from models import User
3
4 # Tạo người dùng mẫu trong ngữ cảnh ứng dụng
5 with app.app_context():
6  # Tạo người dùng mẫu
7  user = User(username="admin")
8  user.set_password("123456") # Đặt mật khẩu cho người dùng
9  db.session.add(user)
10  db.session.commit()
11  print("Người dùng 'admin' đã được tạo thành công!")
```

models.py

```
1 from datetime import datetime
2 from werkzeug.security import generate_password_hash, check_password_hash
   from flask_sqlalchemy import SQLAlchemy
   from flask_login import UserMixin
   db = SQLAlchemy()
   class User(db.Model, UserMixin):
        __tablename__ = 'users'
        id = db.Column(db.Integer, primary_key=True)
        username = db.Column(db.String(50), unique=True, nullable=False)
        email = db.Column(db.String(120), unique=True, nullable=False)
        password = db.Column(db.String(200), nullable=False)
        created_at = db.Column(db.DateTime, default=datetime.utcnow)
        def set_password(self, password):
            self.password = generate_password_hash(password)
        def check_password(self, password):
            return check_password_hash(self.password, password)
   class Book(db.Model):
        __tablename__ = 'books'
        id = db.Column(db.Integer, primary_key=True)
        title = db.Column(db.String(120), nullable=False)
        author = db.Column(db.String(120), nullable=False)
        published_year = db.Column(db.Integer, nullable=False)
        description = db.Column(db.Text)
30 class Rental(db.Model):
        __tablename__ = 'rentals'
        id = db.Column(db.Integer, primary_key=True)
        user_id = db.Column(db.Integer, db.ForeignKey('users.id'), nullable=False)
        book_id = db.Column(db.Integer, db.ForeignKey('books.id'), nullable=False)
        rental_date = db.Column(db.DateTime, default=datetime.utcnow)
        return_date = db.Column(db.DateTime, nullable=True)
        user = db.relationship('User', backref='rentals')
        book = db.relationship('Book', backref='rentals')
```

Index.html

```
dy {
    background-image: url('https://img.icons8.com/?size=100&id=11176&format=png&color=000000');
    background-repeat: no-repeat;
    background-position: center;
    background-sizie: 150px 150px;
    background-color: #f5f5f5;
lackground-c
}
</style>
</head>
<body class="bg-light">
            And the control of th
```

```
<div class="container mt-5">
                Add Book Form
            <h2 class="text-center mb-3">Thêm Sách</h2>
           <form action="{{ url_for('add_book') }}" method="POST" class="border p-4 bg-white rounded">
               <div class="form-group"</pre>
                   < clabel for="title">Tên sách:</label>
<input type="text" name="title" class="form-control" required>
                   <label for="author">Tác giả:</label>
<input type="text" name="author" class="form-control" required>
                   <label for="published_year">Năm xuất bản:</label>
                    <input type="number" name="published_year" class="form-control" required>
                   <label for="description">Mô tả:</label>
                   <textarea name="description" class="form-control"></textarea>
               <button type="submit" class="btn btn-success btn-block">Thêm Sách</button>
           <h2 class="text-center mb-3 mt-5">Danh Sách Sách</h2>
            {% for book in books %}
                   <form action="{{ url_for('delete_book', id=book.id) }}" method="POST" class="d-inline">
                           <a href="{{ url_for('edit_book', id=book.id) }}" class="btn btn-primary btn-sm ml-2">S\check{va}</a>
           <div class="container":
               LibraryApp © 2024 - All Rights Reserved
       <!-- Bootstrap JS and jQuery -->
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```

edit.html

login.html

```
| I <|DOCTYPE html>
| shtml lang="en">
| shead>
| shead>
| shead>
| shead>
| stite-logins/title>
| stite-logins/title>
| shody class="bg-light">
| shody class="bg-light">
| shody class="container mt-5">
| shody class="text-center">Dāng Nhập</h2>
| stone for="nation="fe" người dùng: sheat|
| shody class="form-group">
| shody class="btn btn-primary btn-block">
| shody class="form-control" required>
| shody class="btn btn-primary btn-block">
| shody class="btn btn-primary bt
```

my_rentals.html

register.html

```
. .
             <!DOCTYPE html>
             <html lang="en">
                            <meta charset="UTF-8">
                            <title>Register</title>
                          \verb|\clink| href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"| rel="stylesheet"> | com/bootstrap/4.5.2/css/bootstrap.min.css"| rel="stylesheet"> | com/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/bootstrap/4.5.2/css/boot
            <body class="bg-light">
                                         <h2 class="text-center">Đăng Ký</h2>
                                        <form action="{{ url_for('register') }}" method="POST" class="border p-4 bg-white rounded">
                                                                  <label for="username">Tên người dùng:</label>
                                                                    <input type="text" name="username" class="form-control" required>
                                                      <div class="form-group">
                                                                    <label for="email">Email:</label>
                                                                    <input type="email" name="email" class="form-control" required>
                                             </div>
<div class="form-group">
                                                                   <label for="password">Mật khẩu:</label>
                                                                    <input type="password" name="password" class="form-control" required>
                                                     <button type="submit" class="btn btn-primary btn-block">Đăng Ký</button>
```

rent_books.html