

Тестове завдання: Розробка веб-застосунку для обліку книг у бібліотеці

Мета завдання: Оцінити навички кандидата в галузі веб-розробки на Python за допомогою фреймворку Django.

Завдання:

Ви маєте створити веб-додаток для обліку книг у бібліотеці. Застосунок повинен забезпечувати наступний функціонал:

- Додавання нової книги із зазначенням назви, автора, жанру та року випуску.
- Перегляд списку всіх книг з можливістю їх редагування та видалення.
- Пошук книг за різними параметрами (назва, автор, жанр, рік випуску).
- Позначка книги як прочитана.
- Фільтрування книг за статусом прочитання (всі прочитані, непрочитані).

Вимоги до технологій:

- Використовуйте фреймворк Django для створення веб-застосунку.
- Використовуйте БД для зберігання інформації про книги.
- Використовуйте систему шаблонів Django для відображення даних.
- Реалізуйте механізм аутентифікації (реєстрація/вхід до системи).

Критерії оцінки:

- Структура проекту та відповідність принципам Django.
- Робота з БД, включаючи створення, читання, оновлення та видалення записів.
- Використання системи шаблонів Django для відображення інформації.
- Коректна обробка форм та валідація даних.
- Чистота та структурованість коду.
- Можливість пошуку та фільтрації книг.

Додаткові бали будуть присуджені за:

- Використання AJAX для оновлення даних без перезавантаження сторінки.
- Додавання додаткових функціональних можливостей (наприклад, сортування, статистика тощо).

Зауваження:

- Код має бути написаний в єдиному стилі та містити коментарі до основних частин.
- Результат виконання завдання має бути наданий у вигляді архіву з вихідним кодом, а також інструкцією з розгортання програми.

Додаткові завдання:

Додайте у ваш веб-додаток розділ статистики з використанням графіків для візуалізації інформації про книги в бібліотеці. Реалізуйте такі пункти:

Статистика за жанрами:

Створіть графік, що відображатиме розподіл книг за жанрами.

Надайте можливість вибору часового інтервалу для аналізу (наприклад, за роками чи десятиліттями).

Статистика за авторами:

Створіть графік, який показує кількість книг для кожного автора.

Реалізуйте візуальний елемент (наприклад, кругова діаграма), який відображає частку кожного автора від загальної кількості книг.

Статистика за прочитаними книжками:

Додати візуалізацію про те, скільки книг прочитано, а скільки залишилося.

Можете використовувати горизонтальний графік з поділом на "прочитано" та "залишилося".

Загальна статистика:

Створіть діаграму, яка представляє загальну кількість книг у бібліотеці.

Додати додаткові елементи, такі як середній вік книг, найпопулярніший жанр тощо.

Вимоги:

Використовуйте бібліотеку для візуалізації даних (наприклад, Matplotlib, Plotly, Django Chartit).

Можливість динамічного оновлення графіків без перезавантаження сторінки (AJAX).

Зробіть інтерфейс інтуїтивно зрозумілим та зручним для користувача.

Додаткові бали будуть присуджені за оригінальність підходу, креативність у виборі типів графіків та інформативність наданих даних.

Додаткове завдання з тестування:

Додайте до вашого проекту модульні тести для основних компонентів вашої веб-додатку. Реалізуйте такі тести:

Тести моделей:

Напишіть тести для перевірки коректності створення, читання, оновлення та видалення записів у базі даних для моделей, що використовуються у вашому проєкті (наприклад, модель для зберігання інформації про книги).

Тести відображень (views):

Переконайтеся, що дані на сторінці коректно відображають дані.

Напишіть тести для перевірки обробки різних сценаріїв, таких як додавання, редагування, видалення книг та інші дії.

Тести форм:

Напишіть тести для форм, що використовуються у вашому додатку.

Перевірте, чи форми валідують дані правильно і створюють або оновлюють записи у базі даних.

Тести API (якщо застосовано):

Якщо ваш застосунок надає API, напишіть тести, щоб перевірити коректність роботи API методів.

Перевірте, чи дані правильно повертаються та обробляються при запитах до API.

Тести візуалізації:

Якщо у вашому проєкті використовуються графіки або інші елементи візуалізації, напишіть тести для перевірки їхнього коректного відображення.

Тести безпеки:

Перевірте, що ваша програма захищена від поширених загроз (наприклад, CSRF-атак, ін'єкцій).

Вимоги:

- Використовуйте фреймворк для тестування Django.
- Окремі тестові випадки кожному з функціональних частин.
- Покриття коду тестами має бути максимальним.

Додаткові бали будуть присуджені за:

- Використання моків та фікстур для ізоляції тестів.
- Автоматизація запуску тестів та аналіз покриття.

Успіхів у виконанні ТЗ!