

СТВОРЕННЯ WEB-ОРІЄНТОВАНОГО ДОДАТКУ ДЛЯ ОБЛІКУ ЗАСОБАМИ FLASK

Житомирський державний університет імені Івана Франка, Україна

У сучасному світі обробка і надання в зрозумілому для користувача вигляді великих обсягів даних є одним з найнеобхідніших напрямків в прикладному програмуванні. Для вирішення цих та багатьох інших пов'язаних завдань можна виділити цілий ряд рішень. Але одним з найпопулярніших, мабуть, є подання необхідних даних у вигляді реляційних таблиць. І багато пов'язаних таблиць називають (реляційною) базою даних. Хоча найпоширенішими і використовуваними системами управління базами даних є клієнт-серверні, для великого кола завдань можна і потрібно використовувати вбудовані СУБД. Вони займають менше дискового простору, використовують менше машинних ресурсів, а головне, вони можуть бути розгорнуті локально без використання парадигми «клієнт-сервер», хоча при їх використанні і є ряд обмежень.

Завданням дослідження було створення кросплатформного графічного інтерфейсу користувача для зручної роботи з вбудованою базою даних на прикладі СУБД SQLite. А саме, інтерфейсу, що не потребує установки і працює на ряді операційних систем. Повністю розроблений інтерфейс міг би знайти застосування в особистій сфері використання на домашніх комп'ютерах (наприклад телефонна книга, фотоальбом, книга особистих доходів/витрат) або в середовищі малого бізнесу (облік відвідуваності працівниками, корпоративний щоденник) з огляду зручності і компактності.

Було розроблено WEB-сайт, на якому є можливість вести звітність виконання робіт, облік картриджів та створення запитів на перевірку унікальності тексту. В ролі основної мови програмування використаємо Python, базу даних – SQLite.

Реалізовано такий функціонал:

- реєстрація користувача та авторизація для ідентифікації;
- можливість відновлення забутого паролю;
- сторінка звітності про виконання робіт та рейтинг по кількості звітів;
- сторінка обліку картриджів;
- сторінка відправки запиту на перевірку унікальності тексту;
- сторінка управління користувачами;
- сторінка управління запитами на перевірку унікальності тексту;
- сторінка управління картриджами.

Для зручності написання коду і його підтримки, поділили проект на окремі файли:

- **main.py** – файл із основними функціями додатку, і основною логікою роботи;
- **config.py** – файл із налаштуваннями;
- **funcs.py** – файл з часто використовуваними функціями.

Для опису загального принципу роботи, взято випадок коли користувач має намір додати файл на перевірку унікальності тексту:

- користувач проходить реєстрацію, для подальшого моніторингу запиту та отримання результату.
- запит записується у базу з усіма потрібними даними.
- адміністратор проводить перевірку та оголошує результат.
- запит переміщується з вкладки «В процесі» у вкладку «Перевірені» або ж «Відхилені».

- при натисканні кнопки «Завантажити довідку» – формується довідка з шаблону .docx за допомогою бібліотеки docxtr1 для подальшого друку.
- користувач заходить у вкладку «Мої запити» та бачить результат.

ІНВЦ Головна Інформація Сервіси Контакти Махенько Ярослав Адмін-панель Вихід

Перевірка на антиплагіат

Ваші відправки на антиплагіат

Назва:

Автор:

Науковий керівник:

Тип документа:

Довідка:

Файл: Файл не вибран

Перевірка на антиплагіат
При заповненні форми дотримуйтесь наступних вимог:

1. В полі "Автор" вкажіть прізвище та ім'я у форматі "Петров П."
2. Вкажіть прізвище та ім'я наукового керівника в такому ж форматі (якщо є).
3. Оберіть тип документа, який відповідає вашій роботі.
4. Завантажте файл допустимого розширення.

Допустимі розширення: .docx, .doc, .rtf, .pdf.

В разі порушення цих вимог - заявка на перевірку буде ВІДХИЛЕНА, внаслідок чого - потрібно буде зробити повторну відправку.

Рис. 6. Форма створення запиту на перевірку тексту на унікальність

Під час виконання проекту нами було вивчено середовище програмування PyCharm. Досліджено його функціонал та основні сфери застосування, зв'язок із зовнішньою базою даних SQLite та можливістю підключення до більш вагомих проектів (модульність). Серед вагомих переваг зазначимо можливість масштабування.

Також у відповідності до завдань дослідження було організовано роботу з базою даних через графічний інтерфейс користувача. Тобто фактично було вирішено задачу побудови запитів до бази даних на основі заданих шаблонів по заповненим користувачем полях, використовуючи HTML5, CSS, JS та Python.

Доповнення функціональності інтерфейсу і додавання нових можливостей (видалення записів, редагування, тощо) реалізовано частково, так як додавання нових елементів керування потребує правки вихідного коду.

Було реалізовано реєстрацію та авторизацію користувачів з використанням форм, а також можливістю авторизації через соціальну мережу Facebook.

Проект було адаптовано для потреб ІНВЦ Житомирського державного університету імені Івана Франка та введено в експлуатацію в березні 2020 року. (Доступно лише з внутрішньої мережі університету за посиланням: <https://portal.zu.edu.ua/db/>)

В ході експлуатації було виявлено ряд помилок, які були усунені, так як програмний продукт не було протестовано відповідно до вимог щодо тестування програмних продуктів.

На даний момент було створено 27 облікових записів, 19 зовнішніх та 8 співробітників інформаційно навчально-видавничого центру. Також було внесено понад 500 запитів до бази даних користувачами та працівниками.

Проект розвивається шляхом додавання нового функціоналу та нових сервісів.

Перспективи подальшої роботи є створення електронного документообігу для потреб здобувачів вищої освіти університету.

Література.

1. Python [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Python>.

2. SQLite [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/SQLite>.

3. Руководство SQLite [Електронний ресурс] – Режим доступу до ресурсу: <https://proglib.io/p/sqlite-tutorial>.

Рецензент: **Кривонос О.М.**, к.п.н., доцент кафедри прикладної математики та інформатики Житомирського державного університету імені Івана Франка