Пояснительная записка

В данной пояснительной записке приведен краткий обзор функциональности сайта, построенного на основе фреймворка Flask и содержащего следующие маршруты:

1. @app.route('/') и @app.route('/index'): отображают домашнюю страницу сайта.
2. @app.route('/login', methods=['POST', 'GET']): позволяет пользователям войти в свою учетную запись через форму входа. Метод GET используется для отображения страницы входа, а метод POST для проверки учетных данных пользователя.
3. @app.route('/subjects/<subject>/stats'): позволяет просмотреть статистику по конкретному предмету, передаваемому в качестве параметра в URL.
4. @app.route('/add\_result/<subject>', methods=['POST', 'GET']): позволяет добавлять результаты тестирования пользователя для конкретного предмета, передаваемого в качестве параметра в URL. Метод GET используется для отображения формы для добавления результатов, а метод POST для сохранения результатов в базе данных.
5. @app.route('/register', methods=['GET', 'POST']): позволяет пользователям зарегистрироваться на сайте через форму регистрации. Метод GET используется для отображения страницы регистрации, а метод POST для сохранения данных пользователя в базе данных.
6. @app.route('/settings', methods=['GET', 'POST']): позволяет пользователям изменять настройки своей учетной записи через форму настроек. Метод GET используется для отображения страницы настроек, а метод POST для сохранения изменений в базе данных.
7. @app.route("/user/<id>", methods=["GET", "POST"]): позволяет просмотреть информацию о конкретном пользователе, передаваемом в качестве параметра в URL, а так же принять или отправить заявку учителю(с сохранением в базе данных).
8. @app.route("/subject/<subject>/<user>"): позволяет просмотреть статистику по конкретному предмету, передаваемому в качестве параметра в URL, для конкретного пользователя, передаваемого также в качестве параметра(только для учителя и его учеников).
9. @app.route("/subjects"): позволяет просмотреть список всех предметов, для которых имеется статистика на сайте.
10. @app.route("/all\_samples/<subject>/<int:id>"): позволяет просмотреть все написанные пробники для конкретного предмета, передаваемого в качестве параметра в URL, и конкретного пользователя, передаваемого также в качестве параметра(только для учителя).
11. @app.route("/sample/<id>"): позволяет просмотреть конкретный пробник передаваемый в качестве параметра в URL.
12. @app.route('/add\_teacher/added'): позволяет отобразить ваших учителей из базы данных.
13. @app.route('/add\_teacher'): позволяет отображать список учителей вашей школы.
14. @app.route("/subjects"): позволяет просмотреть список всех предметов, для которых имеется статистика на сайте.

Все данные, введенные пользователями через формы, сохраняются в базе данных, которая используется для хранения информации о пользователях, их результатов тестирования и пробниках. Каждый маршрут обрабатывает запросы соответствующего метода (GET, POST), что позволяет обеспечить корректную обработку пользовательских действий.

Сайт реализован на языке Python с использованием фреймворка Flask и поддерживает множество функций:

1. logout(): это функция, которая обрабатывает запрос на выход из системы пользователя. Она вызывает функцию logout\_user(), которая удаляет пользователя из текущей сессии, а затем перенаправляет пользователя на страницу входа в систему.
2. load\_user(user): это функция, которая используется Flask-Login для загрузки пользователя из базы данных при входе в систему. Она получает идентификатор пользователя (user) и использует его для извлечения соответствующей записи пользователя из базы данных.
3. get\_score\_for\_id(id): это функция, которая извлекает информацию о среднем балле для каждой задачи в экзамене (id). Она извлекает информацию из таблицы TestSeparately в базе данных и вычисляет средний балл для каждой задачи.
4. get\_all\_tasks(subject, ids, sort=False): это функция, которая извлекает информацию о среднем балле для каждой задачи в указанной subject для заданного пользовательского идентификатора ids. Она вычисляет средний балл для каждой задачи, используя информацию из таблицы TestSeparately и возвращает словарь, содержащий средний балл для каждой задачи. Если параметр sort установлен в True, функция возвращает отсортированный словарь с средним баллом для каждой задачи.
5. get\_teachers(school): это функция, которая извлекает информацию о всех учителях, работающих в указанной школе (school). Она извлекает информацию из таблицы User в базе данных и возвращает список всех пользователей с ролью "teacher" и указанной школой.
6. find\_user(id): это функция, которая извлекает информацию о пользователе с заданным идентификатором (id). Она извлекает информацию из таблицы User в базе данных и возвращает запись пользователя с указанным идентификатором.