

Бакалавр компьютерных наук

СМ3035 - Продвинутая веб-разработка

Итоговая курсовая работа: Создание приложения для электронного обучения

Введение

В ходе курса мы разработали несколько приложений с использованием Django. Мы рассмотрели одностраничные приложения, схемы баз данных, проектирование соответствующих моделей, формы и шаблоны, RESTful веб-сервисы, Celery, Django Channels, Web Sockets, аутентификацию и многие другие темы.

В этом задании вам предстоит разработать электронное учебное приложение, используя все знания, полученные на курсе до сих пор.

Это задание составляет 50 % от общей оценки за модуль.

Задание

Реализовать веб-приложение eLearning. Минимальные требования к приложению следующие:

Приложение должно позволять пользователям создавать новые защищенные паролем учетные записи.

У вас должно быть два разных типа пользователей - студенты и преподаватели. Примечание: каждый тип должен иметь различные разрешения, например, преподаватель может иметь доступ к нескольким записям студентов, но это невозможно для учетной записи студента.

Вы должны собирать и хранить соответствующее количество информации о каждом пользователе (например, имя пользователя, настоящее имя, фотография и т. д.).

У каждого пользователя должна быть "домашняя" страница, на которой отображается информация о пользователе и любые другие интересные данные, такие как зарегистрированные курсы, предстоящие сроки и т. д. На ней также должны отображаться обновления статуса пользователя. Эти домашние страницы должны быть доступны и видны другим пользователям.

Студенты должны иметь возможность публиковать обновления статуса на своей

"домашней" странице. Студенты также должны иметь возможность оставлять

отзывы о том или ином курсе. Преподаватели должны иметь возможность искать

студентов и других преподавателей.

Преподаватели должны создавать курсы и загружать в них материалы - изображения, pdf-файлы и т. д.

Преподаватели должны просматривать свои курсы и видеть список студентов, записанных на их курс.

Студенты должны видеть список доступных курсов и выбирать те, на которые они хотят записаться.

Кроме того, пользователи должны иметь возможность использовать ту часть приложения, которая использует связь в реальном времени с помощью веб-сокетов. В качестве примера можно привести текстовый чат в реальном времени, общую доску между учениками и учителями и т. д. Другие функции могут также использовать потоковое аудио и передачу файлов. Вы можете использовать веб-сокеты по своему усмотрению, но вы должны включить 1 приложение веб-сокетов в ваше приложение.

Должен быть предусмотрен соответствующий REST-интерфейс для пользовательских данных.

Приложение должно включать в себя функциональность, использующую Web Sockets.

Вы можете разработать макет приложения по своему усмотрению. Каждый функциональный модуль (список студентов, курсы, результаты поиска и т.д.) может быть отдельной страницей или вы можете сделать приложение одностраничным.

Поставляемые материалы

D1. Приложение на django, реализующее веб-приложение для электронного обучения и отвечающее функциональным требованиям. Приложение должно включать несколько пользователей (т.е. студентов и преподавателей) для демонстрации.

D2. Отчет (4000-6000 слов) с описанием приложения и обоснованием его дизайна и функциональности. Отчет должен объяснить, как ваше приложение соответствует требованиям (см. ниже критерии R1-R5). Объясните логику вашего подхода, почему ваш код устроен именно так, как он устроен? Объясните дизайн вашего приложения, охватывая все принятые вами решения по проектированию и реализации. В конце отчета необходимо критически оценить ваше приложение. Обсудите свой дизайн и реализацию, хорошо ли все получилось, какие части приложения могли бы быть лучше? Что бы вы изменили, если бы попытались повторить проект? Отчет также должен включать краткий раздел о том, как запускать модульные тесты. И наконец, включите раздел о том, как распаковать, установить требования и запустить приложение:

- Список всех пакетов и их версий, используемых в вашей реализации.
- Ваша среда разработки, т. е. операционная система и версия python.
- Инструкция по входу на сайт django-admin, т.е. имя пользователя и пароль.
- Учетные данные для входа учителя и учеников
- Укажите, как запускать модульные тесты.

D3: Включите видео с вашим веб-приложением, демонстрирующее и словесно подчеркивающее основные функциональные возможности и ваши достижения. Вы можете загрузить видео в формате .mp4 или воспользоваться альтернативной ссылкой - см. страницу отправки. В видео должно быть показано, как вы:

- устанавливаете приложение, используя файл requirements.txt
- рассказываете о проектировании и нормализации базы данных
- тестируете приложение, выполняя тесты
- запуск приложения и вход в систему - например, отображение обновлений статуса обратной связи для курса
- запустить сервер redis и во втором браузере для входа второго пользователя инициализировать чат между студентами или студентом и преподавателем.

Продолжительность видео не должна превышать 10 минут. Мы рекомендуем записывать видео в формате mp4 с помощью такого программного обеспечения, как OBS.

D4. Бонусные баллы получают те, кто развернет свое приложение с помощью AWS, Digital ocean и т. д. В отчете необходимо указать адрес приложения и данные для входа в систему.

Требования

Мы будем оценивать вашу работу на основе следующих требований и критериев:

R1: Приложение должно реализовать следующее:

- a) Пользователи должны создавать учетные записи
- b) Пользователи могут входить и выходить из системы
- c) Преподаватели могут искать студентов и других преподавателей
- d) Преподаватели могут добавлять новые курсы
- e) Студенты могут записываться на курсы
- f) Студенты могут оставлять отзывы о курсе
- g) Пользователи могут общаться в режиме реального времени
- h) Преподаватели могут удалять / блокировать студентов
- i) Пользователи могут добавлять обновления статуса на свою домашнюю страницу.
- j) Преподаватели могут добавлять файлы (например, учебные материалы) в свою учетную запись, и они будут доступны на главной странице курса.
- k) Когда студент записывается на курс, преподаватель должен быть уведомлен об этом.
- l) Когда в курс добавляются новые материалы, ученик должен быть

уведомлен R2: Приложение также должно использовать:

- a) правильное использование моделей и миграций
- b) правильное использование форм, валидаторов и сериализации
- c) правильное использование django-rest-фреймворка
- d) правильное использование маршрутизации URL
- e) правильное использование модульного тестирования

R3: Приложение должно реализовать соответствующую модель базы данных для моделирования счетов, хранимых данных и отношений между счетами

R4: Приложение должно реализовать соответствующий код для REST-интерфейса, который позволяет пользователям получать доступ к своим данным

R5: В приложении должны быть реализованы соответствующие тесты для кода на стороне сервера

Стиль и техника написания кода

Ваш код должен быть написан в соответствии со следующими рекомендациями по стилю и технике:

C1: Код четко организован в соответствующие файлы (т.е. код представления помещается в соответствующий файл `view.py` или `api.py`, модели помещаются в соответствующий файл `models.py`)

C2: Соответствующие комментарии включены, чтобы обеспечить ясность и читабельность кода

C3: Код изложен четко, с последовательными отступами, в идеале в соответствии со стандартом `python pep8`

C4: Код разбит на соответствующие функции с четкой, ограниченной целью

C5: Функции, классы и переменные имеют осмысленные имена, с последовательным стилем именования

C6: Приведены соответствующие тесты, покрывающие функциональность API.

Представление

Вы должны написать краткий отчет, записать видеодемонстрацию и предоставить исходный код. Отчет должен содержать следующие элементы и информацию:

S1: Документы D1, сжатые в стандартный формат .ZIP. S2:

Документы D2 и D4 в формате .PDF.

S3: Документ D3 в формате .mp4.

S4: Поставка D3 - альтернативная ссылка - используйте YouTube или аналогичный ресурс и отправьте ссылку. **Убедитесь, что ваше видео не попало в список.**

Критерии оценки

Приложение будет оцениваться по тому, насколько оно технически корректно и реализует API в соответствии с требованиями. Код должен быть понятным и легким для понимания. Приложение должно быть хорошо организовано - например, в нем должны правильно использоваться модели, API, представления и файлы сериализатора. Хорошее приложение будет включать набор тестов, которые гарантируют, что приложение правильно реализует описанный API.

Подробности смотрите в отдельном файле рубрики.