

Техническое задание

В качестве домашнего задания предлагается выполнить проект «**Вопросы и Ответы**». Этот сервис позволит пользователям Интернета задавать вопросы и получать на них ответы. Возможности комментирования и голосования формируют сообщество и позволяет пользователям активно помогать другим. В качестве образца реализации рекомендуется использовать (<http://stackoverflow.com>)

Используемые технологии

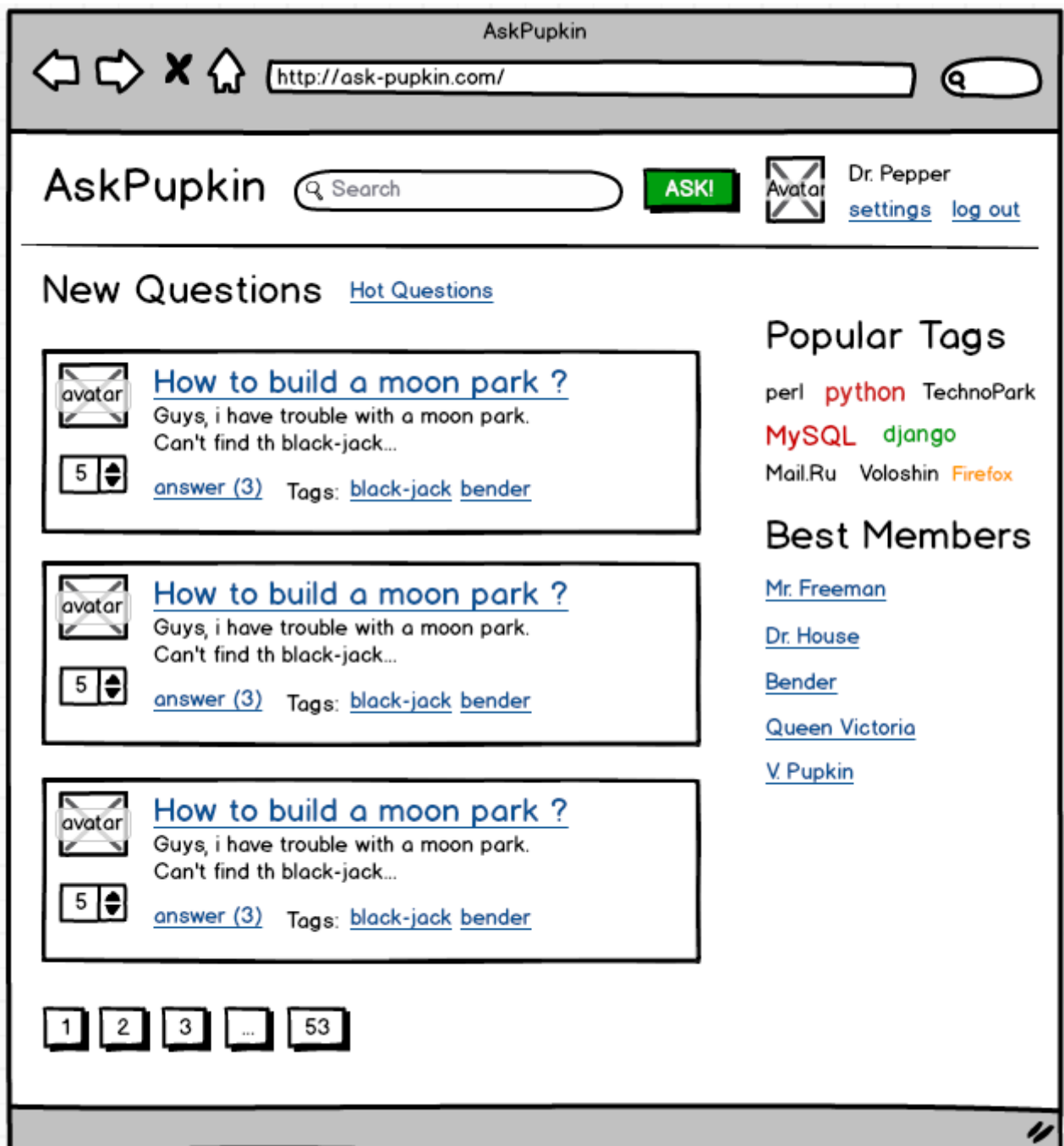
Код приложения пишется на **python + django**, приложение запускается под управлением сервера **gunicorn**. База данных – **mysql**. Для отдачи статики используется **nginx**. Для доставки real-time сообщений **nginx + mod_push**. Для кеширования данных - **memcached**. Верстка выполняется с использованием **Twitter Bootstrap**. Взаимодействие интерфейса с пользователем обеспечивается **javascript/jquery**. Для авторизации и хранения пользователей можно использовать приложение **django.contrib.auth**.

Основные сущности

1. Пользователь – электронная почта, никнейм, пароль, аватарка, дата регистрации, рейтинг.
2. Вопрос – заголовок, содержание, автор, дата создания, тэги, рейтинг.
3. Ответ – содержание, автор, дата написания, флаг правильного ответа, рейтинг.
4. Тэг - слово тэга.

Основные страницы и формы

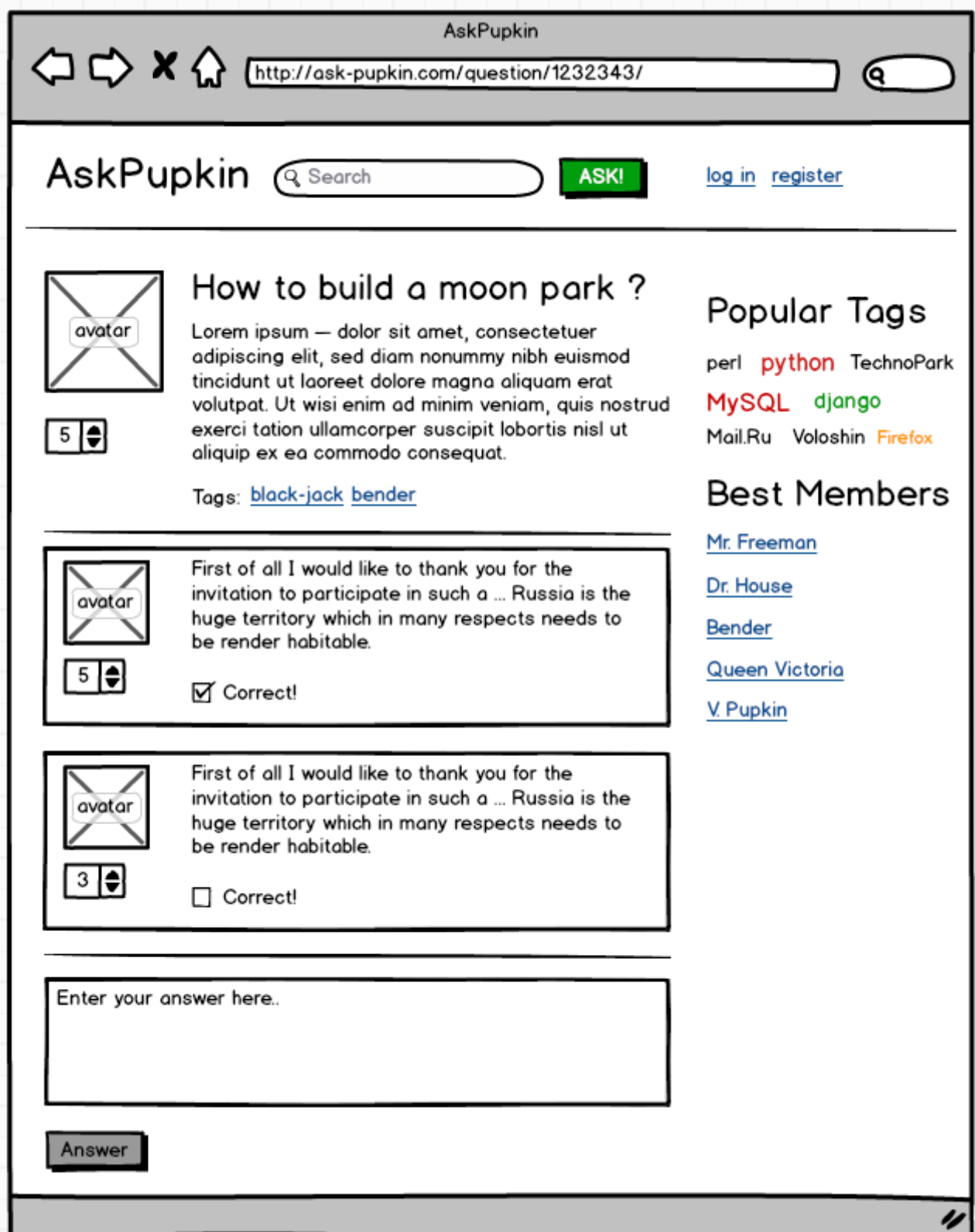
1. Листинг вопросов с пагинацией по 20 вопросов на странице. Необходимо реализовать сортировку по дате добавления и рейтингу (2 вида сортировки). В шапке сайта находятся: логотип, поисковая строка (для быстрого поиска), кнопка задать вопрос (доступна только авторизованным). В правой части шапки - юзерблок. Для авторизованного пользователя юзерблок содержит его ник, аватарку, ссылки на “выход” и на страницу с его профилем. Для неавторизованных - ссылки “войти” и “регистрация”. В правой колонке - информационные блоки “Популярные тэги” и “Лучшие пользователи” (описание ниже). Во всех листингах присутвуют кнопки “лайк/дизлайк”, позволяющие менять рейтинг вопроса.



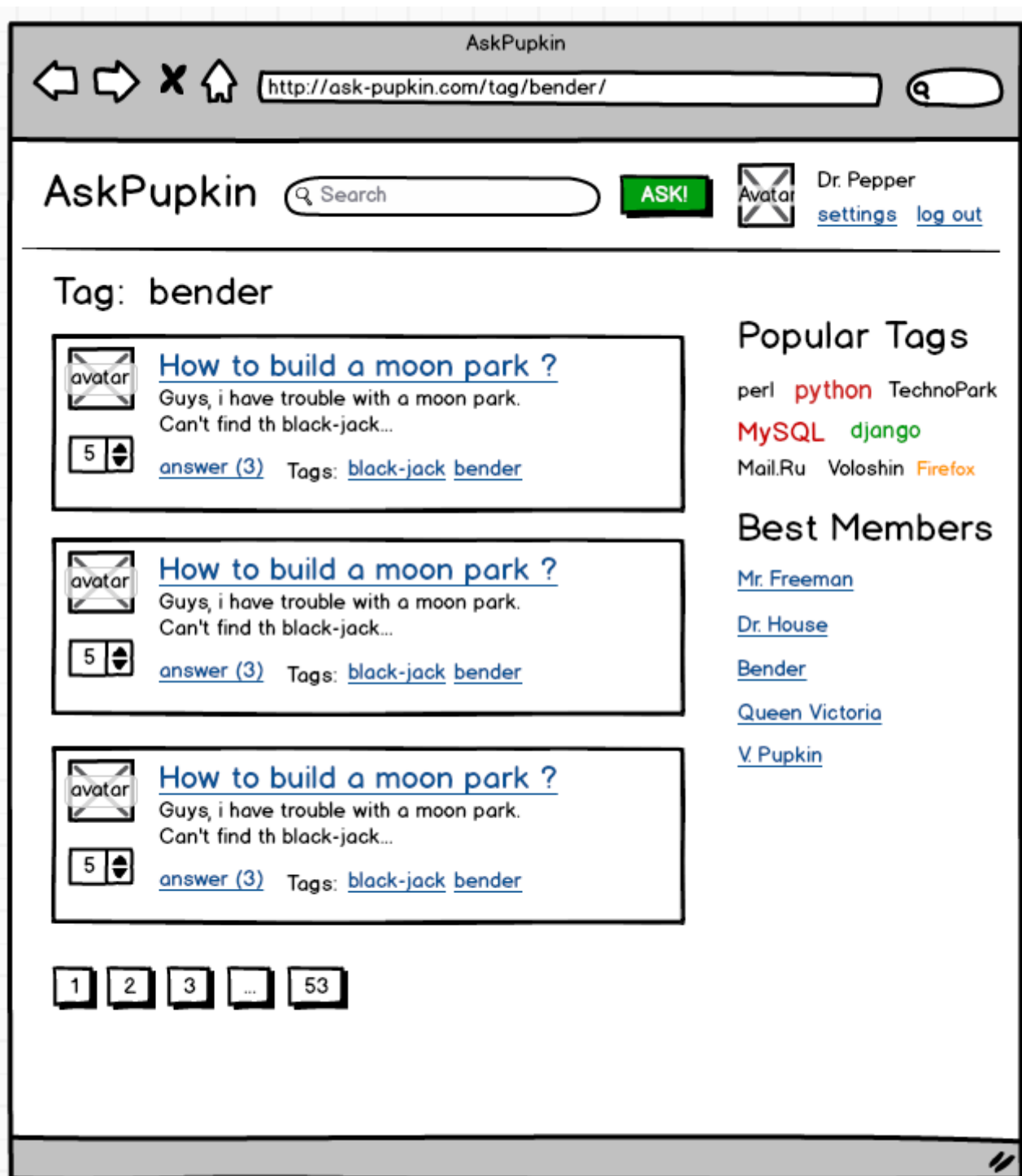
2. Страница добавления вопроса (можно сделать оверлеем). Доступна только для авторизованных пользователей. В форма вводится заголовок, текст вопроса и **тэги, через запятую**. С вопросом может быть связано не более 3 тегов. Для подсказки при выборе тега можно использовать готовый jquery плагин. Готовые django приложения для тэгов использовать запрещается. При обработке формы обязательно проверка валидности данных. Если вопрос успешно добавлен - пользователя перебрасывает на страницу вопроса, **если возникли ошибки - их нужно отобразить в форме**.

The screenshot shows a web browser window with the address bar displaying 'http://ask-pupkin.com/ask/'. The page title is 'AskPupkin'. Below the title is a search bar with a magnifying glass icon and the word 'Search', followed by a green 'ASK!' button. To the right of the search bar are links for 'log in' and 'register'. The main content area is titled 'New Question'. It contains three input fields: 'Title' with the text 'How to build a moon park ?', 'Text' with the text 'Really, how ?' and 'Have no idea about it', and 'Tags' with the text 'moon, park, puzzle'. Below the 'Tags' field is another green 'ASK!' button. On the right side of the page, there is a sidebar with two sections. The first section is 'Popular Tags' with a list of tags: 'perl', 'python', 'TechnoPark', 'MySQL', 'django', 'Mail.Ru', 'Voloshin', and 'Firefox'. The second section is 'Best Members' with a list of names: 'Mr. Freeman', 'Dr. House', 'Bender', 'Queen Victoria', and 'V. Pupkin'. All names in the 'Best Members' section are underlined, suggesting they are links.

3. Страница вопроса со списком ответов. На странице вопроса можно добавить ответ. Ответы сортируются по рейтингу и дате добавления при равном рейтинге. Ответы разбиваются по 30 штук на странице. Форма добавления ответа находится на странице вопроса. Отображается **только для авторизованных пользователей**. После добавления ответа, **автор вопроса должен получить email** с уведомлением от новом ответе. В этом письме должна быть ссылка для перехода на страницу вопроса. Автор вопроса может пометить **один из ответов как правильный**. Пользователи могут голосовать за вопросы и ответы с помощью лайков «+» или «-». Один пользователь может голосовать за 1 вопрос и ответ **только 1 раз**, однако может отменить свой выбор или переголосовать неограниченное число раз.



4. Листинг вопросов по тэгу. На этой странице выводятся все вопросы содержащие некоторый тэг. Сортировка по рейтингу вопроса. Пагинация по 20 вопросов. Пользователи попадают на эту страницу кликая по одному из тэгов в описании вопроса.



5. Страница пользователя содержит его настройки - email, nick и аватарку. Каждый пользователь может смотреть только свою страницу. У пользователя должна быть возможность изменить email, nick и аватарку.

The screenshot shows a web browser window with the address bar displaying `http://ask-pupkin.com/settings/`. The page title is "AskPupkin". The navigation bar includes a search box with the text "Search", a green "ASK!" button, and a user profile section for "Dr. Pepper" with an "Avatar" icon and links for "settings" and "log out".

Settings: Dr. Pepper

The settings form contains the following fields and controls:

- Login:** A text input field containing the value `dr_pepper`.
- Email:** A text input field containing the value `dr.pepper@mail.ru`.
- NickName:** A text input field containing the value `Dr. Perpper`.
- Upload avatar:** A section containing a placeholder image with an "X" and the text "avatar", a text input field with the value `c:\avatar.jpg`, a "choose" button, and a "Save" button.

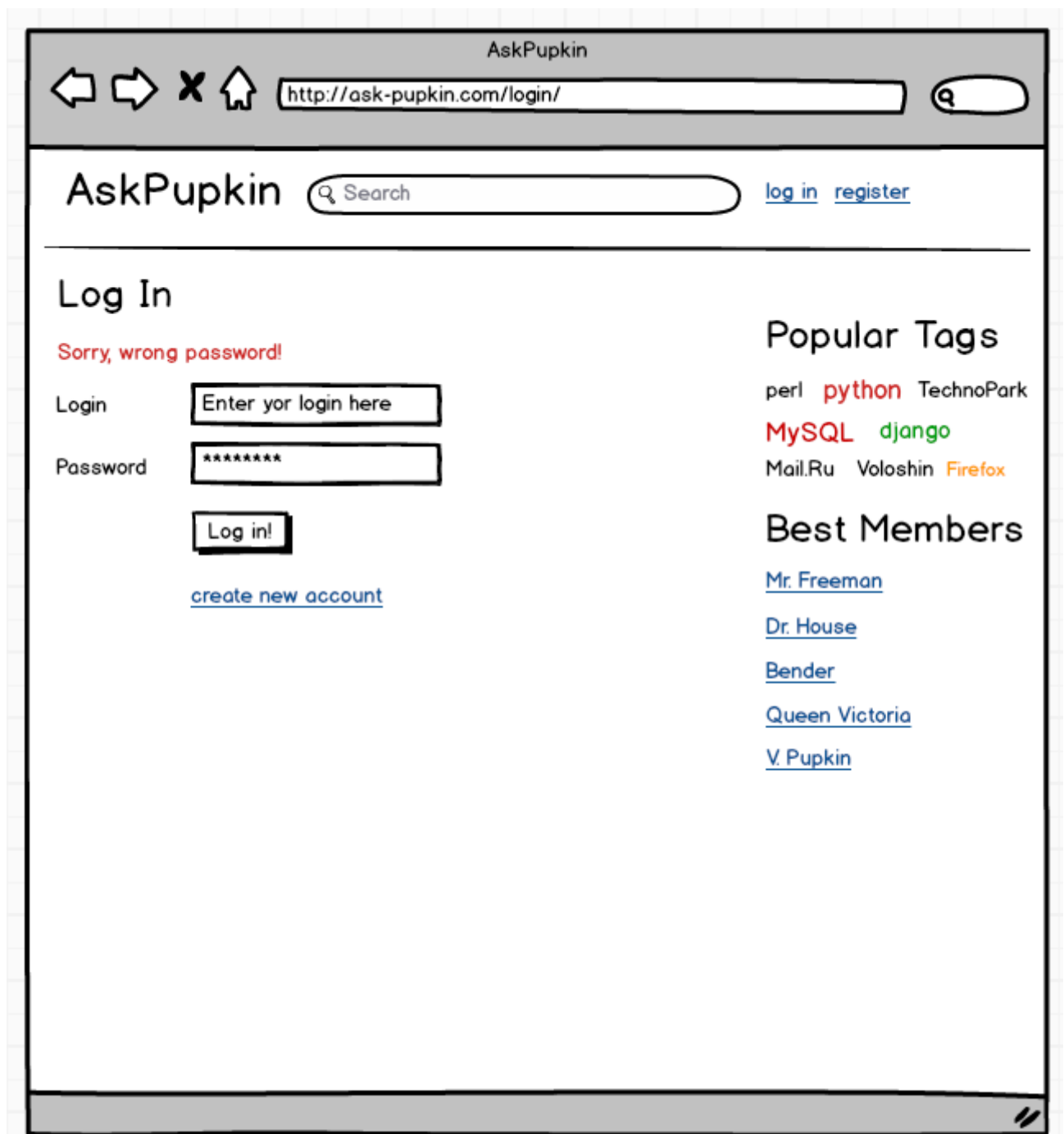
Popular Tags

A list of tags: `perl`, `python`, `TechnoPark`, `MySQL`, `django`, `Mail.Ru`, `Voloshin`, and `Firefox`.

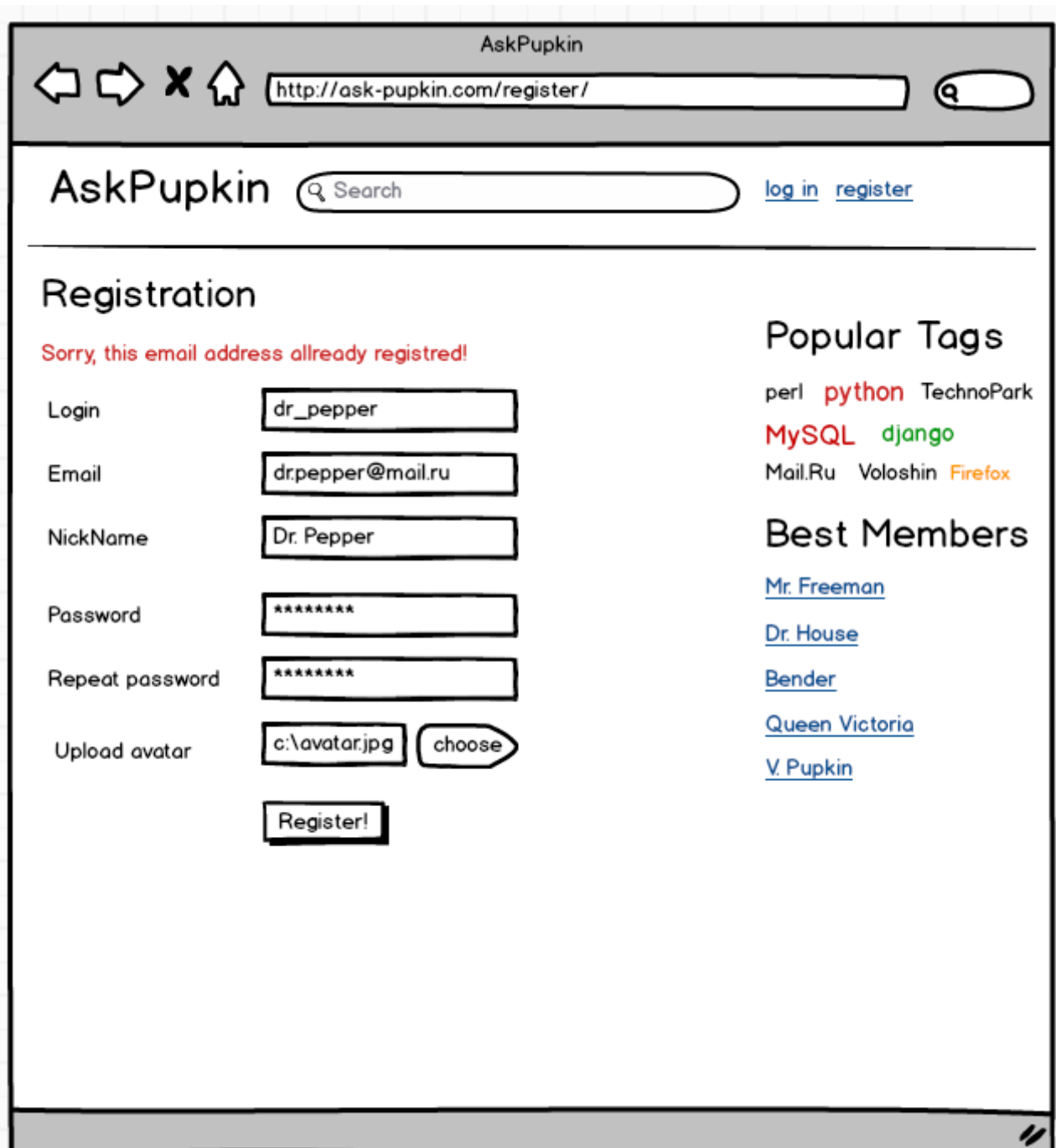
Best Members

A list of member names with blue underlined links: `Mr. Freeman`, `Dr. House`, `Bender`, `Queen Victoria`, and `V. Pupkin`.

6. Форма авторизации. Состоит из поля логин и пароль. Дополнительно есть ссылка на форму регистрации. При успешной авторизации пользователь перебрасывается на исходную страницу, при неуспешной авторизации - ему показывается ошибка. Для авторизованных пользователей вместо этой формы должна показываться кнопка "Выйти".



7. Страница регистрации. Любой пользователь может зарегистрироваться на сайте, заполнив форму с электронной почтой, никнеймом, аватаркой и паролем. **Аватарка загружается на сервер и отображается рядом с вопросами и ответами пользователя.**



8. Блок популярных тэгов. В правой колонке сайта находится облако из 20 наиболее популярных тегов. Популярными считаются те тэги, которые были использованы в наибольшем количестве вопросов. Генерация этого блока занимает много времени, поэтому этот блок необходимо генерировать **в фоне, с помощью cron скрипта**.
9. Блок лучших пользователи (недели). В блок лучшие пользователи попадают 10 авторов задавших самые популярные вопросы или ответы за последнюю неделю. Т.е. вопросы и ответы, созданные за последнюю неделю сортируем по рейтингу. Выбираем топ N, их авторы и будут "лучшими". Генерация этого блока занимает много времени, поэтому блок нужно подготавливать **в фоне, с помощью cron скрипта**.

Требования к проекту

1. Структура проекта должна быть понятна пользователям. Переходы по страницам осуществляются по ссылкам. Обработка форм должна осуществляться с редиректом.
2. Код проекта должен быть аккуратным и без дублирования. Наличие больших повторяющихся фрагментов кода или шаблонов могут быть причиной снижения баллов.
3. Верстка проекта должна быть выполнена с помощью css фреймворка Twitter Bootstrap.

4. Код приложения должен быть чувствителен к входным данным и выдавать соответствующие коды и тексты ошибок. Сообщение пользователям «Вопрос добавлен», «Вопрос не был добавлен, потому что» выводятся в оверлее. Ответ сервера с кодом 500 может быть причиной снижения баллов.
5. Время генерации страницы не должно зависеть от объема данных в базе.
6. Страницы проекты не должны отдаваться более **1 секунды**.

Требования к объему данных

- Пользователи > **10 000**.
- Вопросы > **100 000**.
- Ответы > **1 000 000**.
- Тэги > **10 000**.
- Оценки пользователей > **2 000 000**.

Дополнительные задания.

1. Real-time нотификации, уведомление о новых ответах при просмотре вопроса (**20 баллов**). С использованием long-polling или web sockets.

Оценка и разбаловка

Пороговые баллы

- **0-39** — неудовлетворительно.
- **40-59** — удовлетворительно.
- **60-79** — хорошо.
- **80-100** — отлично.

ДЗ 1 “Статическая верстка” - 16 баллов.

Верстка общего вида (layout) страницы - **4**:

- общий вид: 2 колонки, header, footer - **1**;
- правая колонка - **1**;
- блок авторизованного юзера - **1**;
- поисковая строка и логотип - **1**.

Верстка листинга вопросов - **3**:

- общий вид (паддинги, аватарка) - **1**;
- кнопки лайков - **1**;
- тэги, счетчики ответов, остальное - **1**.

Верстка страницы вопроса - **3**:

- общий вид - **1**;
- чекбокс “правильный ответ”, кнопки лайков - **1**;
- форма ответа - **1**.

Верстка формы добавления вопроса - **3**:

- общий вид - **2**;
- сообщения об ошибках - **1**.

Верстка форм логина и регистрации - **3**:

- общий вид - **2**;
- аватарка и сообщения об ошибках - **1**.

ДЗ 2 “Настройка серверов” - 14 баллов.

Настройка nginx для отдачи статического контента - 5:

- общее - 3;
- локейшен /upload/ приоритетнее статики - 1;
- проксирование - 1.

Настройка gunicorn для запуска wsgi приложений - 2:

- общее - 2.

Создание hello-world WSGI приложения - 2:

- собственно hello-world - 1;
- отображение списка GET, POST и других параметров - 1.

Создание django проекта - 2:

- создание структуры проекта - 1;
- создание hello-world view - 1.

Оценка производительности nginx и gunicorn - 3:

- отдача статики - 1;
- запуск WSGI - 1;
- запуск WSGI с проксированием через nginx - 1.

ДЗ 3 “Обработка HTTP запросов” - 14 баллов.

Создать views и шаблоны для основных страниц - 6:

- главная (список новых вопросов) - 1;
- страница вопроса (список ответов) - 1;
- страница добавления вопроса - 1;
- форма регистрации - 1;
- форма входа - 1;
- форма добавления вопроса - 1.

Создать urls.py для всех страниц - 4:

- Собственно urls.py - 2;
- Именованные маршруты (во всех шаблонах) - 2.

Постраничное отображение - 4:

- функция пагинации - 1;
- шаблон для отрисовки пагинатора - 2;
- корректная обработка “неправильных” параметров - 3.

ДЗ 4 “Работа с базой данных” - 14 баллов.

Проектирование модели - 5:

- правильные адекватные типы данных и внешние ключи - 1;
- своя модель пользователя - 1;
- таблицы тэгов, лайков - 1;
- query-sets для типовых выборок: новые вопросы, популярные, по тэгу - 2.

Наполнение базы тестовыми данными - 3:

- наполнение данными - 2;
- использование django management commands - 1.

Отображение списка вопросов - 3:

- список новых вопросов - 1;

- список популярных - **1**;
- список вопросов по тэгу - **1**.

Отображение страницы вопроса - **3**:

- общее - **3**.

ДЗ 5 “Авторизация и обработка форм” - 15 баллов.

Вход на сайт - **3**:

- общее - **3**;
- возврат на исходную страницу - **1**;
- отображение ошибок - **1**.

Регистрация на сайте - **3**:

- общее - **2**;
- отображение ошибок - **1**.

Выход с сайта - **1**.

Отображение текущего пользователя в шапке - **1**.

Добавление вопроса - **4**:

- общее - **1**;
- добавление тэгов - **1**;
- отображение ошибок - **1**;
- редирект на страницу вопроса - **1**.

Добавление ответа - **2**:

- общее - **1**;
- редирект на добавленный ответ - **1**.

Проверка метода запроса **1**.

ДЗ 6 “Изображения и обработка AJAX запросов” - 11 баллов.

Загрузка и отображение аватарок пользователей - **4**:

- заливка картинок в uploads - **2**;
- отображение картинок из uploads - **2**.

Страница редактирования профиля - **2**:

- общее - **1**;
- отображение ошибок - **1**.

Лайки вопросов и ответов - **2**:

- общее - **1**;
- ajax - **1**.

Отметка “правильный ответ” - **2**:

- общее - **1**;
- ajax - **1**.

Проверка авторизации, csrf, метода запроса, авторства - **1**.

ДЗ 7 “Дополнительные функции” - 16 баллов.

Real-time сообщения - **8**:

- настройка nginx+mod_push - **2**;
- подключение к mod_push с помощью jQuery - **3**;
- отправка сообщений в mod_push - **3**.

Блок популярные тэги - **4**:

- правильный расчет тэгов - **2**;
- предварительный расчет по cron - **2** (просто кеширование - **1**).

Блок лучшие пользователи - **4**:

- правильный расчет пользователей - **2**;
- предварительный расчет по cron - **2** (просто кеширование - **1**).