Доска объявлений

Содержание

- 1. Концепция проекта
- 2. Используемые технологии
- 3. Основные сущности
- 4. Uses Case
- 5. Матрица возможностей
- 6. Список задач

1. Концепция проекта

Проект представляет собой доску объявлений, на которой можно продавать товары и услуги.

Типы пользователей на сайте:

- **Гость** - неавторизованный пользователь, который может просматривать объявления и делать поиск по фильтрам и регистрироваться в личном кабинете, а также проходить авторизацию.
- **Пользователь** - авторизованный пользователь, который может работать с объявлениями (CRUD) и редактировать информацию о себе.
- **Модератор** - авторизованный пользователь, который может модерировать объявления и допускать/не допускать к публикации на сайте.
- **Администратор** - авторизованный пользователь, который может делать, все, что остальные роли, и редактировать справочники (категории, города и тд).

Для каждой роли(кроме гостя) необходимо разработать личный кабинет, в который будет попадать пользователь после авторизации, и в котором будут отображаться только доступные для данной роли разделы. Для некоторых ролей могут быть доступны одинаковые страницы, решается настройкой прав доступа.

Перед тем, как разместить объявление в поиске для всех, пользователю необходимо зарегистрироваться, подтвердить почту, создать объявление, отправить его модератору на проверку. После успешной модерации объявления, его можно найти в поиске и оно доступно абсолютно всем гостям сайта.

2. Используемые технологии

Цель данного проекта: получить широкий набор компетенций по разработке проекта на Symfony Framework, который максимально приближен к коммерческому проекту.

При разработке использовать следующие технологии:

- PHP 7+/Python 3
- PHP: Laravel, Symfony, Python: Django
- PostgreSQL
- Docker, Docker-compose
- RabbitMQ (обязательно поднять админ панель для просмотра состояния очереди)
- Recaptcha (опционально)
- Elastic Search
- Redis (опционально, дял хранения сессий)
- Bootstrap (можно взять готовую тему, дизайн не важен, но верстка должна быть нормальной), Vue.js (опционально)

3. Основные сущности

3.1. Пользователь:

- Имя
- Фамилия
- Отчество (опционально)
- Роль
- email (уникальное поле)
- Номер телефона (уникальное поле)
- Когда удобно принимать звонки (текстом)
- Статус пользователя:
  - заблокирован
  - активен
  - ожидает активации email

3.2. Объявление:

- Название
- Категория
- Город
- Описание

- Дата публикации
- Стоимость
- Кто продает - пользователь
- Сколько просмотров
- Фотографии
- Статус объявления:
    - черновик (когда его только создал пользователь)
    - на модерации (когда пользователь отправил на модерацию и не имеет доступа к редактированию)
    - отклонено, к доработке (когда модератор не пропустил объявление)
    - снято, продано (когда пользователь удалил объявление)
    - активно (когда модератор пропустил объявление и оно доступно в поиске)

3.3. Категория объявления:

- Название
- Код на английском для красивого url (уникальное поле)
- Описание
- Родительская категория (ссылка на саму себя, чтобы выстроить дерево)
- Порядок сортировки

При заполнении справочника реализовать категории, у которых минимальная глубина была не менее 3-4. Категории можно взять с сайта avito (не обязательно копировать весь список категорий, достаточно 20 шт).

3.4. Справочник городов

- Название города
- Регион

При заполнении справочника взять данные из sql скрипта country_city_dump.sql

3.5. Справочник регионов

- Название региона

При заполнении справочника взять данные из sql скрипта country_city_dump.sql

3.6. Запись о модерации объявления

- дата модерации
- объявление

- модератор - пользователь
- решение:
  - опубликовать
  - отправить на доработку
- Причина отклонения к доработке (причину выводим пользователю)

4. Uses Case

В данном разделе описаны основные варианты использования системы, но не все (нужно дополнить, если будут спорные моменты).

4.1. Регистрация нового пользователя

- Гость переходит на страницу с формой регистрации и заполняет необходимые для регистрации пользователя данные. Дополнительно на форме может быть recaptcha, которую нужно пройти. На форме есть js валидация полей при помощи js плагина.
- При отправке формы и обработке запроса на сервере, происходит также валидация формы. Если данные заполнены неверно, то пользователю выводится сообщение об ошибке.
- Если валидация прошла успешно, необходимо проверить нет ли пользователя с таким номером телефона или с таким email.
- Если все ок, то в базе данных создается пользователь в статусе "Ожидает активации email" и на его почту отправляются письмо с ссылкой на страницу активации.
- Пользователю, выводится сообщение, что необходимо перейти по ссылке из письма, чтобы завершить регистрацию в системе.

4.2. Авторизация

- Авторизоваться может только тот пользователь, который имеет статус "Активный".
- При авторизации также нужно выводить captcha и проверять ее (отображается только после 3 неверных попыток авторизации, привязыватсья к логину).

4.3. Просмотр объявлений

- Просматривать объявления можно всем, но только если объявление в статусе "Активно". Иначе, можно просматривать объявление только модератору, администратору и тому, кто его создал.
- Только на подробной странице объявления можно посмотреть контакты для связи с продавцом.

4.4. Поиск объявлений

- Поиск осуществляется только по тем объявлениям, которые находятся в статусе "Активно".
- Фильтрация объявлений возможна последующим параметрам:
  - город
  - область/регион
  - категория
  - стоимость от
  - стоимость до
  - текст (полнотекстовый поиск по содержимому и по названиям)

При отправке формы с фильтрами, отображаются товары с пагинацией.

4.5. Создание объявления

- Авторизованный пользователь заполняет информацию о товаре или услуге, загружает до 10 изображений. Выбирает среди них главное изображение. Анимированные изображения загружать запрещено.
- После отправки формы и валидации формы, в базе создается запись о новом объявлении в статусе "Черновик".
- Продавцом в данном объявлении указывается текущий пользователь
- Это объявление видно только в ЛК пользователя, который его создал.
- Пока объявление в статусе "Черновик", пользователь может его редактировать и менять любую информацию.

4.6. Отправка объявления на модерацию

- Пользователь может отправить объявление на модерацию, тогда у него скрывается возможность редактировать объявление и оно переходит в статус "На модерации".

4.7. Модерация объявления

- В ЛК модератора должна быть специальная страница, которая позволяет просматривать объявления в статусе "На модерации", просматривать их на сайте, как они будут смотреться. И также видит все истории модерации объявления, чтобы проверить учтены ли исправления.
- Модератор может вынести два решения: отклонить и опубликовать (создается сущность "Запись о модерации объявления")
  - Если объявление опубликовано, то оно переходит в статус "Активно" и дата публикации ставится текущей датой.

- Если объявление отклонено к доработке, то Модератор должен указать причину отклонения. Тогда объявление меняется в статус "Отклонено к доработке".

4.8. Редактирование объявления

- редактировать объявления может только пользователь, который его создал.
- Пользователь может редактировать только то объявление, которое находится в статусе "Черновик" или "Отклонено к доработке". После редактирования, пользователь может снова отправить на модерацию.
- Если объявление было отклонено, то пользователю выводится сообщение модератора с замечаниями.

5. Матрица возможностей

Возможность/Роль	Гость	Пользователь	Модератор	Администратор
4.1. Регистрация нового пользователя	+	-	-	+
4.2. Авторизация	-	+	+	+
4.3. Просмотр объявлений	+	+	+	+
4.4. Поиск объявлений	+	+	+	+
4.5. Создание объявления	-	+	-	-
4.6. Отправка объявления на модерацию	-	+	-	-
4.7. Модерация	-	-	+	+
4.8. Редактирование объявления	-	+	-	-

6. Список задач

6.1. Создание каркаса проекта

- Создать docker-compose с необходимыми сервисами и каркасом фреймворка внутри.
- Описать в файле `Readme.md` инструкцию по разворачиванию окружения в Docker

- В результате должен получиться репозиторий, где в корне лежит docker-compose и при поднятии всех сервисов по запросу http://bulletin-board.test/ открывалась приветственная страница фреймворка.

Полезные ссылки:

- Docker
- Docker-compose
- Alpine Linux in docker
- Пример проекта с docker-compose

6.2. Проектирование базы данных и сущностей приложения

- В результате выполнения данной задачи, должны быть разработаны миграции базы данных и сущности Entity для работы с ними
- Нет требования четко определить бд на данном этапе, в процессе разработки она может дорабатываться. Но основной каркас должен быть реализован на данном этапе.
- Таблицы базы данных должны быть минимум в 3 нормальной форме и иметь все необходимые индексы и поля.
- Все модели должны обладать автоматически заполняемыми полями `created_at` и `updated_at`

Полезные ссылки:

- Нормализация отношений. Шесть нормальных форм
- Уровни изоляции транзакций с примерами на PostgreSQL
- Индексы в MySQL
- Выбор типов данных в MySQL
- Explain

6.3. Разработка авторизации, регистрации и восстановления доступов

- Реализовать авторизацию в системе по связке email + password
- Проверять капчу на форме от google reCAPTCHA
- Реализовать страницу активации учетной записи
- Реализовать отправку писем с активационной ссылкой на email пользователя
- Реализовать возможность восстановления пароля

Полезные ссылки:

- Google reCAPTCHA
- Как надо хешировать пароли и как не надо

- Protecting passwords with Argon2 in PHP 7.2
- Using Argon2 with Django

6.4. Настройка списка ролей пользователей

- Реализовать роли пользователей в системе, разработав для каждой роли свою страницу в лк, которая доступна только для нее.
- После авторизации пользователя, переводить его на страницу ЛК, которая создана под эту роль.
- Описать в `Readme.md` инструкцию, как добавить администратора в систему

Полезные ссылки:

- Аутентификация и авторизация в микросервисных приложениях

6.5. Разработка ЛК и страниц заглушек для каждой роли

- Разработать для каждой роли страницы в ЛК, пока без вывода данных, но с четким описанием на какой странице какая функциональность будет.
- Закрыть эти страницы правами с указанием ролей, которые имеют доступ к ним.

6.6. Реализовать возможность создания объявления

- Реализовать страницу добавления нового объявления в ЛК пользователя
- Реализовать проверку прав при попытке открыть страницу с неверной ролью.
- Реализовать сохранение фотографий в нужную директорию (анимированные фото нельзя, а также слишком мелкие < 300x300px).
- Если фотография по одной из сторон боле 1500px, то нужно уменьшить фото с сохранением пропорций
- Все формы должны быть защищены CSRF

Полезные ссылки:

- OWASP CSRF
- Методы защиты от CSRF-атаки

6.7. Реализовать просмотр объявления

- Реализовать просмотр объявлений с учетом уровней доступов:
  - Объявление в статусе "Активно" могут видеть все, включая гостей
  - Объявление в статусе "Черновик", "Отклонено к доработке", "Снято, продано" могут видеть только пользователи, которые создали это объявление (остальным 404)

    - Объявление в статусе "На модерации" могут просматривать автор объявления и модератор
- Если объявление в статусе "Отклонено к доработке", то также должна выводиться история модераций этого объявления, чтобы можно было проследить что должен был доработать пользователь
- Для описания конечного автомата состояний объявления можно использовать The Workflow Component

Полезные ссылки:

- Для вывода фото использовать Карусель

6.8. Реализовать отправку объявления на модерацию

- Реализовать необходимые проверки для отправки объявления на модерацию
- Реализовать отправку уведомления на почту модератору, что есть новое объявление для модерации
- Если объявление отправлено на модерацию, пользователь не может его редактировать

6.9. Реализовать просмотр объявлений к модерации в ЛК модератора и Админа (отказ с причиной или модерацию)

- Реализовать в ЛК модератора страницу со списком объявлений на модерацию (сортировка по дате оправки на модерацию по возрастанию)
- Реализовать возможность отправить объявление на доработку, указав что именно нужно дрaдоработать текстом.
- Реализовать возможность опубликовать объявление

6.10. Реализовать редактирование объявления

- Редактировать объявление может только его автор и только в статусах "Черновик", "Отклонено к доработке".
- После редактирования пользователь также должен иметь возможность отправить на модерацию объявление.

6.11. Реализовать удаление, снятие объявления

- Если пользователь что-то продал, то он может снять с публикации объявление
- Объявление из бд не удаляется, оно просто становится неактивным и его никто не может просматривать, кроме автора и модераторов

6.12. Реализовать CRUD пользователей в ЛК админа

- Реализовать CRUD пользователей в ЛК администратора
- После создания пользователя, на почту отправляется активационное письмо со специальной ссылкой для активации
- При переходе по специальной ссылке активации, пользователю предлагают задать пароль.
- Если установить пароль и снова перейти по активационной ссылке, должно выводиться сообщение, что данная ссылка устарела.

6.13. Реализовать CRUD справочников в ЛК админа

Реализовать CRUD справочников в ЛК админа:

- справочник Регионов
- справочник Городов
- справочник Категорий объявлений (в виде дерева)

6.14. Реализовать поиск объявлений

- Поиск осуществляется только по тем объявлениям, которые находятся в статусе "Активно".
- Фильтрация объявлений возможна последующим параметрам:
  - область/регион
  - город (доступен только после выбора области, отображает только города из выбранной области)
  - категория
  - стоимость от
  - стоимость до
  - текст (полнотекстовый поиск по содержимому и по названиям)
- При отправке формы с фильтрами, отображаются товары с пагинацией.
- Для полнотекстового поиска использовать ElasticSearch
- Реализовать консольный воркер, который индексирует данные в ElasticSearch
- Реализовать команду-seed, которая при запуске заполняет базу рандомными объявлениями (использовать faker), чтобы индексатор их отправил в ElasticSearch

Полезные ссылки:

- Основы Elasticsearch
- Эффективный поиск на сайте с помощью Elasticsearch
- Ранжирование объявлений на основе машинного обучения и ElasticSearch
- Почему ivi перешел со Sphinx на Elasticsearch
- Полнотекстовый поиск в веб-проектах: Sphinx, Apache Lucene, Xapian

6.15. Реализовать отправку писем через очередь сообщений

Отправка писем через SMTP - долгое занятие, особенно, если SMTP внешний и нужно дождаться его ответа. Тем самым запросы будут идти долго.
Чтобы ускорить ответ сервера, мы будем отправлять задание на отправку сообщения в очередь RabbitMQ, которую будет разбирать консольная команда и обрабатывать.
Обработка будет осуществялться по следующему алгоритму:

- взять задание из очереди
- через шаблонизатор отрендерить нужный шаблон письма
- отправить письмо

Для отладки отправки сообщений используйте сервис mailtrap.io

Полезные ссылки:

- RabbitMQ Tutorials
- mailtrap.io
- php-amqplib/php-amqplib
- Pika

6.16. Покрыть тестами основную бизнес-логику

Необходимо покрыть тестами функциональность добавления и модерации объявлений в личном кабинете для разных пользователей.

Полезные ссылки:

- PHP
  - Testing: Getting Started
  - Codeception
  - Codeception for Symfony
  - Тестирование с Codeception для чайников: 3 вида тестов
  - Тесты на Codeception для PHP-бэкендов (Lamoda)
- Python
  - Introducing automated testing
  - Тестирование приложений Django