Веб-дизайнер

Сервис бронирования железнодорожных билетов

Создать дизайн-макеты основных страниц онлайн-сервиса для бронирования железнодорожных билетов (как бы выглядел e-kvytok.com.ua в идеальном мире?)

Набор макетов произволен, но он должен обеспечивать выполнение следующих задач пользователя:

- поиск билетов (выбор даты, пункта направления, пункта назначения)
- выбор поезда, вагона, места
- оплата бронирования

Работы будут оцениваться по 100-бальной системе по трем критериям:

- **Удобство использования** (здесь под удобством использования понимается степень эффективности, трудоемкости и удовлетворенности, с которыми продукт может быть использован определенными пользователями (в нашем случае — пассажиры Укрзализныци) при определенном контексте использования (бронирование билетов) для достижения определенных целей/мотивов (спокойно и без очереди выбрать наиболее подходящий по времени и качеству поезд, подходящие вагон и место, легко оплатить билет))
- **Аккуратность, внимание к деталям**.
- **Визуальная привлекательность** (композиция, контраст, цветовая схема, качество графики)

Дополнительно, по желанию членов жюри, могут быть начислены бонусные баллы:

- за качественную работу с текстом
- за досрочное выполнение работы

Результат конкурсной работы —любой набор изображений в формате PNG, послойные макеты не нужны.
Пожалуйста, не упаковывайте свои файлы в RAR-архивы, это вводит членов жюри из душевного равновесия.

Backend Developer

3-

Завдання: **Написати генератор статичних сайтів**

Опис завдання **:**
Написати генератор сайтів зі статичним вмістом з підтримкою однієї з наступних мов розмітки:

- Markdown
- ReStructured Text
- Wiki

Генератор повинен аналізувати вміст папки (назва папки - назва сайту), знаходити HTML-шаблони та текстові файли у одній з вищенаведених розміток для вставки у шаблони.

Мінімальний набір функціоналу повинен включати:

- Парсинг шаблонів HTML з placeholder-ами для вставки динамічних даних
- Парсинг файлів розмітки
- Генерація вихідних HTML-файлів
- Можливість вивантаження готових сайтів на github з їх подальшим відображенням

Критерії оцінки:

- правильність роботи парсерів
- правильність підстановки динамічного контенту за різних HTML-шаблонів
- захищеність генерованих сайтів незалежно від поданого динамічного вмісту
- код та архітектура додатку

Бонуси

- можливість вставки зображень та інших типів медіаданих у тому ж синтаксисі
- вивантаження сайту на Amazon S3
- можливість вибору стилів оформлення вихідного сайту

Правила

Використання готових рішень, а також рішень на основі готових – суворо заборонено.

Вибір серверної мови та використовуваних бібліотек – на вибір учасника.

Рішення оцінюється по 100 бальній шкалі. Бонуси надаються за рішенням журі.

3+

Задание: **Написать скрипт синхронизации контактов из Zimbra в Constant Contact и произвести рассылку по ним**

Описание задания **:**
Constant Contact (http://constantcontact.com) - распространенный на Западе инструмент проведения маркетинговых компаний.

Очень часто в качестве почтового комплекса используется Zimbra (http://zimbra.com). Задача состоит в изучении API двух систем и написания скрипта-синхронизатора контактов с последующей рассылкой.

Зимбру можно установить к себе, если у вас Linux, либо использовать готовую которая находится по адресу - https://mail.jampo.com.ua (для получения аккаунта нужно зарегистрироваться на http://jampo.com.ua).

Конечным результатом должен быть скрипт (консольный по-возможности), в настройках которого должны быть данные для доступа к обеим аккаунтам, который сможет:

- импортировать контакты из вашего GMail-аккаунта в Zimbra-аккаунт
- сихнронизировать после этого контакты в Zimbra с аккаунтом на Constant Contact

Критерии оценки:
- код и архитектура приложения
- правильность работы скрипта

Бонусы
- возможность двухсторонней синхронизации (Zimbra -> Constant Contact, Constant Contact -> Zimbra)
- CLI-настройки скрипта и расширенные возможности по запуску
- осуществить рассылку по базе полученных в результате контактов в Constant Contact (текст письма на тему проведения финала UWC) с возможностью возобновления в случае остановки скрипта вручную либо разрыва связи
- использование SOAP для работы с Zimbra

Правила

Использование готовых решений, а так же решений на основе готовых - строго запрещено. Использовать фреймворки при этом можно.

Выбор серверного языка и используемых библиотек - на выбор исполнителя.

Решение оценивается по 100 бальной шкале. Бонусы присваиваются на усмотрение жюри

Frontend Developer

3-/3+

Задание: Сверстать макет с максимальной поддержкой браузеров. Возможно использование graceful degradation

Описание задания :
Макет доступен для скачивания по адресу – http://uwc.org.ua/public/files/uwc.rar .
Все части предполагающие взаимодействие с пользователем (прокручивание картинок-контента, тултипы-подсказки) должны быть реализованы с помощью javascript.

Обратите внимание на элементы подразумевающие слайдинг (перемещение, прокручивание) картинок. А также переключение видимых блоков (табы), и тултипы.

Не стесняйтесь задать вопрос членам жюри! Футер должен быть прижат к низу, если контента недостаточно чтобы заполнить экран по высоте.

Критерии оценки:

- Соответствие стандартам, валидность; современность решения
- Работа в современных браузерах (последние версии Chrome, Firefox; IE9),
- соответствие макету
- Работа в старых браузерах (IE6-8) (не обязательно идентично, но как
- минимум доступно — допускается применение progressive enhancement /
- graceful degradation)
- Адаптивность (поведение в экранах разного размера, включая планшеты
- и мобильные устройства) — продумывание изменения дизайна при этом
- является частью задания

- Поддерживаемость кода, удобство внедрения, элегантность решений,
- форматирование
- Доступность (accessibility), семантичность
- Быстродействие, компактность решения

Правила

Выбор используемых библиотек - на усмотрение исполнителя.

Решение оценивается по 100 бальной шкале.

Architect

Задание

Описать архитетурное решение для видеохостинга аналогичного YouTube. Основные требования

1. Загрузка/выгрузка видео значительным числом одновременно авторизированных пользователей
2. Минимизация времени обработки файла необходимого для згрузки/выгрузки
3. Возможность горизотально масштабировать систему (использование кластеризации)
4. Возможность вертикально масштабировать систему (параллельные вычисления, обработка информации в cloud или grid)
5. Организация хранилища для файлов большого объема, а также организация доступа к файлам внутри этого хранилища (поиск, обновление, удаление)

Значительным бонусом будет детализация архитектуры онлайн видео стриминга, однако основной задачей по прежнему является создать архитектуру для хранения видео файлов значительного объема и организации доступа к этой информации.

Критерии приемки

1. Диаграммы
2. Выбор хранилища для файлов большого объема и его интеграция с системой
3. Масшабируемость системы
4. Устойчивость системы к нагрузкам

Бонусы

1. Детатизация архитектуры видео стриминга
2. Архитектурное решение показа рекламы (пероллы, построллы и т.д.) при стримминге

Tester

Задача: **тестирование** http://uwc.org.ua/
Эта задача потребует от вас навыков планирования и оптимизации своей работы. У вас будет только 6 часов на всё задание. Вам необходимо распределить это время эффективно и быть ориентированным на результат. Результат и понимание качества продукта - самое важное в нашей работе, по этому сконцентрируйтесь на достижении этой цели!

Протестировать сайт http://uwc.org.ua/:

- Создать список того какие тестовые активности будут выполнены;
- Создать тест кейсы;
- Записать найденные баги;
- Создать предложения.
- Создать отчет о проделанной работе.

Тестовые активности:
Все, которые вы, как QA Engineer посчитаете необходимыми для определения качества тестируемого продукта.

Предоставление результатов и оценочные критерии:

1. **Тестовые активности** - соотношение выполненного к запланированному, наличие всех необходимых и важных активностей;
2. **Тест кейсы** - количество, качество, покрытие, удобность использования и апдейта, полученные результаты, количество пройденных кейсов.

3. **Найденные баги** - количество и качество описания, критичность.
4. **Отчет о проделанной работе** - результаты проделанной работы, оценка качества проекта, метрики, статистика, выводы.

Дополнительные **баллы могут быть получены за:**

1. Комментарии и предложения - смысл, описание, количество, ценность для продукта;
2. Использование автоматизированной тестирования
3. Тестирование на Mac и/или мобильных платформах

Команда

Завдання: **Графічне резюме користувача**

Опис завдання **:**
Написати сервіс побудови графічного резюме користувача з використанням LinkedIn API.

Мінімальний набір функціоналу повинен включати:

- Автентифікація через LinkedIn
- Можливість вибору користувачем даних для резюме
- Відображення базових даних, професійного та освітнього досвіду тощо у графічному timeline-режимі чи іншому зручному для користувача графічному способі подання інформації.
- Попередній перегляд вихідного резюме.

Критерії оцінки:

- код та архітектура додатку
- дизайн
- безпека додатку (XSS, CSRF, SQL injection тощо)
- можливість редагування імпортованих даних

Бонуси

- генерація PDF-файлу з резюме

- можливість підключення іншим сервісів в якості вхідних даних (Facebook, Google+ тощо)
- API для сервіса (REST, SOAP, XML-RPC тощо)

Правила

Використання готових рішень, а також рішень на основі готових – суворо заборонено.

Вибір серверної мови та використовуваних бібліотек – на вибір учасника.

Рішення оцінюється по 100 бальній шкалі. Бонуси надаються за рішенням журі.