**2023. ИТИС. ОРИС.**

**СЕМЕСТРОВАЯ РАБОТА 2 (ВЕБ-ПРОЕКТ)**

**2 КУРС 3 СЕМЕСТР**

Работа должна быть организована в группах по 4 человека (роли распределяете сами). Если в какую-то группу не хватит людей, то придётся организовать одну из 2-5 человек, с некоторым ограничением или наоборот расширением функционала по согласованию со мной.

**Основная задача:**

Написать веб-приложение для некоторой реальной области. Тематика и дизайн сайта будет выдано вам случайным образом.

По выбранным шаблонам и тематикам сайта нужно будет утвердить ТЗ.

**Технологический стек:**

* **Web-приложение:** .NET MVC (движок Razor), EntityFramework, Identity, SignalR, AutoMapper.
* **База данных:** Postgresql/ MS SQL, Redis,
* **Сервер:** Kestrel
* **Фронт:** html5, css3 (с библиотекой bootstrap или альтернативной), js с библиотекой jquery (прочие библиотеки или фреймворки по согласованию с преподавателем)
* **Трекер задач:** Jira, YouTrack, Github или другие

**Ключевые моменты:**

* Изначально необходимо описать основные функциональные требования к приложению. Эти требования следует хорошо продумать и согласовать преподавателем (пример требований приложен). Необходимо также продумать, какой функционал будет возможно расширен в будущем, т.е. у вас должны быть точки расширения для вашего приложения, при реализации которых не должна страдать архитектура.
* Архитектуру приложения следует продумывать исходя из функциональных требований. Она должна быть спроектирована таким образом, чтобы была возможность легкого изменения хранилище данных, при этом должен минимально затрагиваться остальной код. Кроме того должна быть возможность несложного добавления (с минимумом внесений изменений в код основного приложения) нового варианта пользовательского интерфейса, к примеру, вместо веб-интерфейса использовать мобильный вариант. Естественно, при проектировании классов вы должны придерживаться всех принципов S.O.L.I.D. и минимизировать уровень зависимости вашего кода. Итоговое решение для архитектуры лежит на вашей совести, однако вы должны аргументированно обосновать свой выбор, почему вы выделили те или иные модули или пришли к тому или иному варианту реализации.  Предполагается, что ваше веб-приложение (пользовательский интерфейс) будет реализован на основе стандартного ASP.NET Core MVC приложения.
* Обязательно в приложении должно присутствовать разграничение доступа к функциональным модулям в виде ролей пользователей (всё на claims, естественно). Кроме того, касаемо доступа следует реализовать доступ к части функций вашего приложения в виде системы подписок: Пользователь оформляет подписку (виртуально оплачивая счет или распоряжаясь кредитами), получает доступ к ресурсам на определенный по продолжительности период времени. Есть несколько уровней подписок с разной стоимостью (отличаются по срока/функциональности). По истечению срока действия, закрывается доступ до следующего продления (ручного или автоматического).

Система подписок должна подразумевать несложный вариант расширения действия на другие функции веб-приложения. Например, вы реализуете приложение для прослушивания музыки. Ранее вы создали функционал, который заключался в оплате услуги прослушивания треков (при этом функциональность подбора музыки входила в эту же подписку), после чего решили выделить эту функциональность в отдельную платную услугу. Соответственно должна быть реализована возможность для такого выделения, при этом с минимальным изменением кода.

Сами подписки тоже можно менять, добавляя/удаляя сами подписки, изменяя их с добавлением/удалением функциональностей, изменяя сроки и т.д.

* Ваше приложение должно быть безопасным с точки зрения любого пользовательского ввода. Все вводимые данные должны обязательно валидироваться и санитизироваться. Кроме того, в целом вы должны максимально продумать безопасность приложения: двухфакторная авторизация, отсутствие возможности простой смены пароля, адреса электронной почты и т.п.
* Ваша разработка должна быть организована «по-взрослому» - с обязательным использованием репозиториев с регулярными коммитами, изначальным продумыванием структуры БД, архитектуры приложения и только последующей их реализацией, использованием таск-трекера, соблюдением сроков, либо обоснованным их изменением.
* В конце ваше приложения должно быть развернуто либо в Azure, либо на Heroku либо прочий хостинг. При этом опять же все должно быть оптимально, должно быть настроено сжатие ответа, минификация и т.д.

**Разбалловка**

Баллы будут выдаваться на команду, после проверки требований баллы будут делиться на количество членов команды в проекте.

Например: на человека максимально 25 баллов в команде 4 человека, следовательно на команду 100 баллов. При проверке репозитория были найдены нарушения по архитектуре (5 баллов), в таком случае количество 95 баллов будет убавлено на 5 баллов. После проверок технологических требований итоговую сумма команд делется на количество участников 95/4 = 23,75 (округление в меньшую сторону).

Ниже указаны технологические требования к вашему приложению, с разбалловкой.

Технологические требования

|  |  |
| --- | --- |
| Соответствие сайта ТЗ | 1\*a |
| Соответствие сайта макету | 1\*a |
| Соответствие административной части макету | 1\*a |
| Наличие адаптива сайта | 1\*a |
| Функциональность сайта должна быть реализована согласно согласованного ТЗ с преподавателем. | 4\*a |
| Корректное разделение на слои приложения: контроллеров, сервисов, работы с данными и т.д. Выделение бизнес-логики, сервисного кода в отдельные проекты. При некорректной реализации архитектуры баллы будут выставляться не полностью. Если будет реализовано лишь монолитный вариант приложения, то за этот пункт вы получите 0 баллов. | 3\*a |
| Наличие валидаторов данных и форм.  Все данные в модели, естественно должны проверяться корректными, однако кроме стандартных валидаторов вам нужно реализовать 3 различных собственных валидатора для данных. | 1\*a |
| Для списка объектов (в части администрирования портала) сделать постраничный вывод, фильтрацию, сортировки по любому столбцу. Также реализовать возможность поиска объекта. | 1\*a |
| Наличие минимум 3 Ajax-запросов (и соответствующих им контроллеров, возвращающих JSON) | 1\*a |
| Реализовать некий функционал, основанный на веб-сокетах, в частности на SignalR (к примеру, это может быть чат с техподдержкой или некоторые уведомления от администрации веб-приложения, которые пользователь получает онлайн без перезагрузки страницы). | 2\*a |
| Кэш сессий хранить необходимо в Redis | 2\*a |
| Реализовать авторизацию через Google, VK и Yandex | 1\*a |
| Авторизация и права доступа с помощью Identity. Создание 2 ролей с разными правами (например. Администратор и менеджер) с разными уровнями доступа, ну плюс пользовательские роли. Создание собственных политик для некоторых вариантов авторизации. | 2\*a |
| Настройка CI/CD на хостинг | 1\*a |
| Хостинг сайта | 1\*a |
| Наличие логирования приложения и просмотр логов через административную панель (можно использовать NoSQL бд, например MongoDB) | 1\*a |
| Наличие уведомлений на почту о событиях в системе | 1\*a |

Где a – это кол-во человек в команде

Возможные штрафные санкции за несоблюдение и нарушение требований указаны в таблице нижу (список скорее всего будет расширяться). Количество вычитаемых баллов будет варьироваться в зависимости от того, насколько требования были не соблюдены.

Штрафные санкции

|  |  |
| --- | --- |
| Отсутствие полноценного репозитория с регулярными коммитами | -5\*a |
| Отсутствие таск-трекера | -5\*a |
| Невыполнение требования по проектированию БД и архитектуры | -5\*a |
| Регулярные срывы промежуточных сроков | -1\*a |
| Срыв итоговых сроков. | -5\*a |
| Несоблюдения требований безопасности вашего приложения (открытые уязвимости, которые мы проходили на занятиях) | -1\*a |
| Работа не разделена равномерно. (Каждый член команды должен реализовать минимум одну страницу(как фронт так и бекенд часть)) | -7\*a |
| Отсутствие защиты и презентации проекта. Сбои во время защиты и презентации проекта | -5\*a |

## **Примерный план-график работ**

|  |  |
| --- | --- |
| 11.02.2023 - 18.02.2023 | Получение задания. Сбор требований, подготовка ТЗ. Согласование ТЗ. Создание репозитория. |
| 18.02.2023 - 25.02.2023 | Декомпозиция и распределение задач по историям. Начало работы. Перенос верстки. |
| 25.02.2023 - 4.03.2023 | Продолжение разработки, завершение основной верстки, начало тестирования верстки |
| 4.03.2023 - 11.03.2023 | Разработка |
| 11.03.2023 - 18.03.2023 | Разработка |
| 18.03.2023 - 25.03.2023 | Демо-показ. Показ преподавателю промежуточного итога. Актуализация плана. |
| 25.03.2023 - 1.04.2023 | Разработка |
| 1.04.2023 - 8.04.2023 | Демо-показ. Показ преподавателю промежуточного итога. Актуализация плана. |
| 8.04.2023 - 15.04.2023 | Разработка |
| 15.04.2023 - 22.04.2023 | Демо-показ. разработка, настройка ci/cd. |
| 22.04.2023 - 29.04.2023 | Завершение разработки, код-фриз, тестирование |
| 29.04.2023 - 6.05.2023 | Деплой приложения. финальное тестирование, баг-фиксинг |
| 6.05.2023 - 13.05.2023 | Презентация сдача проекта |

В конце семестра вам нужно будет презентовать свой проект, рассказав и показав основные возможности, и технические моменты реализации вашего приложения.

Дедлайн по окончательному коду проекта – 13 мая. До 13 мая вы должны выкатить его на хостинг. К 13 мая у вас должна быть подготовлена презентация. 13 и 20 мая на занятиях планируется защита проектов.

Данные требования могут измениться, но не принципиально (скорее всего, будут просто пополняться). Соответственно возможно изменение в плане весов (баллов), но опять же незначительное.