**2024. ИТИС. ОРИС.**

**СЕМЕСТРОВАЯ РАБОТА**

**(ПРОЕКТ ASP.NET MVC)**

**2 КУРС 4 СЕМЕСТР**

Работа должна быть организована индивидуально с некоторым ограничением или наоборот расширением функционала по согласованию с преподавателем.

**Основная задача:**

Написать веб-приложение для некоторой реальной области, решением которой реально бы пользовались. Тематики и дизайн сайта будет выдано на занятии.

**Ключевые моменты:**

* 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. Эти требования следует хорошо продумать и согласовать со мной (в лице заказчика).
* 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 MVC приложения.
* Обязательно в приложении должно присутствовать разграничение доступа к функциональным модулям в виде ролей пользователей. Пример, система 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 развернут. При этом опять же все должно быть оптимально, должно быть настроено сжатие ответа, минификация и т.д.

**ТРЕБОВАНИЯ И РАЗБАЛЛОВКА**

|  |  |
| --- | --- |
| Архитектура. Корректное разделение на слои приложения: контроллеров, сервисов, работы с данными и т.д. Выделение бизнес-логики, сервисного кода в отдельные проекты. Если будет реализовано лишь монолитный вариант приложения, то за этот пункт вы получите 0 баллов. | 5 |
| Дизайн и функциональность сайта должна быть реализована согласно согласованного ТЗ с преподавателем. Дизайн должен быть адаптивным, легким и быстрым. Дизайн должен соответствовать выданным макетам Стараться минимизировать число перезагрузок страниц. (см. следующий пункт). | 1 |
| Наличие 2-3 (а лучше больше) Ajax-запросов (и соответствующих им контроллеров, возвращающих JSON) | 1 |
| Административная панель. Должна состоять минимум из 6 страниц. (дашборд, управление пользователями, управление контентом сайта) | 1 |
| 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. | 1 |
| Сайт должен состоять минимум из 5 страниц (минимум 3 должны быть не статическими) | 1 |
| Возможность фильтрации данных. (при перезагрузке страницы фильтр должен не сбрасываться) | 1 |
| Наличие валидации при заполнении форм | 1 |
| Наличие пагинации данных (при перезагрузке страницы фильтр должен не сбрасываться) | 1 |
| База данных сайта должна состоять минимум из 6 таблиц (таблицы линковочные и Identity не учитываются). | 1 |
| 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). | 1 |
| Наличие локализации на сайте | 1 |
| Использование дополнительно MongoDB для какой-нибудь подзадачи (при наличии основной реляционной БД MS SQL или PostgreSQL) | 1 |
| Наличие логирования действий на сайте. (какие действия логировать определяет сам студент) | 1 |
| Правильная маршрутизация сайта(404, 403) | 1 |
| Сдача всех этапов согласно календарному плану (так же регулярные коммиты в репозитории) | 5 |
| Наличие минимум одного Middleware | 1 |
| ИТОГО | 25 |

Баллы выставляются следующим образом: Каждый пункт оценивается по реализации по 5 бальной шкале. Далее баллы баллы расчитываются по формуле:

**макс\_баллы\_за\_пункт \* (оценка /5)**

**КАЛЕНДАРНЫЙ ПЛАН**

|  |  |  |
| --- | --- | --- |
| Номер спринта | Дата окончания(сдачи) | Задачи |
| 1 | 9.03.2024 | Черновое ТЗ |
| 2 | 16.03.2024 | Окончательный вариант ТЗ. Создание проекта в репозитории(выдает преподаватель) |
| 3 | 23.03.2024 | Показ перенесенного UI на Razor pages |
| 4 | 30.03.2024 | Добавление функционала на страницы(бэкенд). |
| 5 | 6.04.2024 | Добавление функционала на страницы(бэкенд) |
| 6 | 13.04.2024 | Административная панель. (наличие дашборда, и авторизации) |
| 7 | 20.04.2024 | Административная панель. Добавление CRUD операций контента |
| 8 | 27.04.2024 | Доработки |
| 9 | 4.05.2024 | Демо. Демонстрация проекта. Сбор доработок. |
| 10 | 11.05.2024 | Дедлайн. доделки |
| 11 | 18.05.2024 | Основной день сдачи проекта. Презентация. |
| 12 | 25.05.2024 | Сдача долгов (максимальный балл для получения уменьшается на 25%) |

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 – 18 мая. До 18 мая вы должны выкатить его на хостинг. К 18 мая у вас должна быть подготовлена презентация. 18 и 25 мая на занятиях планируется защита проектов.