1. **Паспорт Образовательной программы**

**«**Разработка интернет-приложений на платформе C# ASP.Net MVC**»**

|  |  |
| --- | --- |
| **Версия программы** | **1** |
| **Дата Версии** | 07**.**10**.**2020 |

1. **Сведения о Провайдере**

|  |  |  |
| --- | --- | --- |
| 1.1 | Провайдер | Федеральное государственное бюджетное образовательное учреждение высшего образования «АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» |
| 1.2 | Логотип образовательной организации | в формате jpeg разрешением не менее 100x100 пикселей |
| 1.3 | Провайдер ИНН | 2225004738 |
| 1.4 | Ответственный за программу ФИО | Козлов Денис Юрьевич |
| 1.5 | Ответственный должность | Заведующий кафедрой информатики |
| 1.6 | Ответственный Телефон | +7(913)2344650 |
| 1.7 | Ответственный Е-mail | dyk.barnaul@gmail.com |

1. **Основные Данные**

|  |  |  |
| --- | --- | --- |
| **№** | **Название** | **Описание** |
| 2.1 | Название программы | Разработка интернет-приложений на платформе C# ASP.Net MVC |
| 2.2 | Ссылка на страницу программы | <https://public.edu.asu.ru/course/view.php?id=879> |
| 2.3 | Формат обучения | Онлайн |
|  | Подтверждение от ОО наличия возможности реализации образовательной программы с применением электронного обучения и (или) дистанционных образовательных технологий с возможностью передачи данных в форме элементов цифрового следа | У ФГБОУ ВО «Алтайский государственный университет» есть возможность реализации образовательной программы с применением электронного обучения и (или) дистанционных образовательных технологий с возможностью передачи данных в форме элементов цифрового следа |
| 2.4 | Уровень сложности | Продвинутый |
| 2.5 | Количество академических часов | **72** |
|  | Практикоориентированный характер образовательной программы: не менее 50 % трудоёмкости учебной деятельности отведено практическим занятиям и (или) выполнению практических заданий в режиме самостоятельной работы (кол-во академических часов) | Да: 24 часа лабораторных работ + 36 часов самостоятельной работы |
| 2.6 | Стоимость обучения одного обучающегося по образовательной программе, а также предоставление ссылок на 3 (три) аналогичные образовательные программы иных организаций, осуществляющих обучение, для оценки объективности стоимости или обоснование уникальности представленной образовательной программы в случае отсутствия аналогичных образовательных программ на рынке образовательных услуг | 15000 р.,  Ссылки на похожие курсы  <https://edu.softline.com/vendors/microsoft/razrabotka-asp-net-mvc-5-veb-prilozhenij/>  <https://www.luxoft-training.ru/kurs/osnovy_raboty_s_platformoy_aspnet_mvc_50.html>  <https://www.specialist.ru/track/t-mstsweb> |
| 2.7 | Минимальное количество человек на курсе | 5 |
| 2.8 | Максимальное количество человек на курсе | 50 |
| 2.9 | Данные о количестве слушателей, ранее успешно прошедших обучение по образовательной программе | При наличии |
| 2.10 | Формы аттестации | Отчет по лабораторной работе,  Защита проекта в форме онлайн-собеседования |
|  | Указание на область реализации компетенций цифровой экономики, к которой в большей степени относится образовательная программа, в соответствии с Перечнем областей | Программирование и создание ИТ-продуктов |

1. **Аннотация программы**

Наиболее полное и содержательное описание программы, которое включает:

1) общую характеристику компетенций, качественное изменение которых осуществляется в результате обучения или которые формируются в результате освоения образовательной программы;

2) описание требований и рекомендаций для обучения по образовательной программе;

3) краткое описание результатов обучения в свободной форме, а также описание востребованности результатов обучения в профессиональной деятельности.

Ограничение по размеру: не менее 1000 символов -?

Компетенция в области Программирование и создание ИТ-продуктов: Способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение - интернет-приложения на платформе .Net. Архитектурный шаблон MVC (model - view - controller) получил большое распространение, особенно в веб-разработке. Присущая ему концепция разделения ответственности между данными, интерфейсом и управляющей бизнес-логикой легко позволяет построить работу над отдельными компонентами, вследствие чего разрабатываемое приложение легко тестировать. Также появляется возможность специализации и распараллеливания задач, что очень удобно для IT-бизнеса. Так, визуальной частью или фронтэндом, может заниматься один разработчик-специалист, другой же сосредоточится на бизнес-логике (или бэкэнде) приложения - контроллере, а работать с данными - третий. Кроме того, разным пользователям с разными ролями можно отображать разные представления (разный фронтэнд), а контроллер (бэкэнд) при этом может оставаться тот же. За счет этого вырастает общая скорость разработки.

Существует возможность разработки веб-приложений на языке программирования C# с использованием платформ ASP.NET MVC и ASP.NET Core MVC в связке с Entity Framework - специальной объектно-ориентированной технологией для работы с данными. В данном курсе рассматривается разработка веб-приложений на языке программирования C# с использованием ASP.NET MVC и Entity Framework. Курс ориентирован на представителей IT-специальностей и всех заинтересованных лиц. Уровень образования: высшее образование, среднее специальное, область профессиональной деятельности – информационные технологии. Требования к уровню подготовки поступающего на обучение – знакомство с языком программирования C#, понимание основных концепций ООП. В результате освоения данного курса обучающиеся должны знать структуру и принципы использования технологии ASP.Net для создания Интернет-приложений. Обучающиеся будут понимать концепцию архитектурного паттерна MVC технологиями, приобретут востребованные в IT-индустрии навыки разработки Интернет приложений на языке C#, использования MS'Visual Studio, Entity Framework, научатся создавать защищенные Интернет-приложения с разграничением доступа в соответствии с политикой ролей.

1. ШАБЛОН ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ (ДПО)

Титульный лист программы

Название организации

Дополнительная профессиональная программа повышения квалификации

« Разработка интернет-приложений на платформе C# ASP.Net MVC »

72 час.

**ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ**

**1.Цель программы**

Компетенция в области Программирование и создание ИТ-продуктов: Способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение - интернет-приложения на платформе .Net. Целью реализации программы повышения квалификации является совершенствование и получение новых компетенций в области веб-разработки, освоение принципов, методов и инструментов создания веб-приложений на основе архитектурного паттерна MVC с использованием платформы .Net и языка программирования C#.

**2.Планируемые результаты обучения:**

2.1.Знание (осведомленность в областях)

2.1.1. структуру и принципы использования технологии ASP.Net для создания Интернет-приложений.

2.2. Умение (способность к деятельности)

2.2.1. разрабатывать Интернет приложения на языке C#, использовать MS'Visual Studio, Entity Framework, создавать защищенные Интернет-приложения с разграничением доступа в соответствии с политикой ролей.

2.3.Навыки (использование конкретных инструментов)

2.3.1 концепцией архитектурного паттерна MVC.

**3.Категория слушателей** (возможно заполнение не всех полей)

Уровень образования: высшее образование, среднее специальное.

* 2. Квалификация
  3. Область профессиональной деятельности – информационные технологии. Представители IT-специальностей и все заинтересованные лица.
  4. Требования к уровню подготовки поступающего на обучение – знакомство с языком программирования C#, понимание основных концепций объектно-ориентированного программирования.

**4.Учебный план программы «…..наименование программы….»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль** | **Всего, час** | **Виды учебных занятий** | | |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | Основы работы с ASP.NET MVC | 12 | 2 | 4 | 6 |
| 2 | Работа с представлениями. HTML-хелперы | 12 | 2 | 4 | 6 |
| 3 | Модели и контроллеры. | 18 | 2 | 6 | 10 |
| 4 | Пагинация. Фильтрация. Валидация | 12 | 2 | 4 | 6 |
| 5 | Веб-приложение MVC с аутентификацией | 16 | 2 | 6 | 8 |
| 6 | Итоговая аттестация (Защита проекта в форме онлайн-собеседования) | 2 | - | 2 | - |
| **Итоговая аттестация** | |  | **Указывается вид (экзамен, зачёт, реферат и т.д.)** | | |
| Итого | | 72 |  | | |

**5.Календарный план-график реализации образовательной** программы

(дата начала обучения – дата завершения обучения) в текущем календарном году, указания на периодичность набора групп (не менее 1 группы в месяц)

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Наименование учебных модулей** | **Трудоёмкость (час)** | **Сроки обучения** |
| **1** | Основы работы с ASP.NET MVC | 12 | 01.11.2020 - 02.11.2020 |
| **2** | Работа с представлениями. HTML-хелперы | 12 | 03.11.2020 - 04.11.2020 |
| 3 | Модели и контроллеры. | 18 | 05.11.2020 - 07.11.2020 |
| 4 | Пагинация. Фильтрация. Валидация | 12 | 08.11.2020 - 09.11.2020 |
| 5 | Веб-приложение MVC с аутентификацией | 16 | 10.11.2020 - 12.11.2020 |
| 6 | Итоговая аттестация (Защита проекта в форме онлайн-собеседования) | 2 | 13.11.2020 |
| **Всего:** | | 72 |  |

**6.Учебно-тематический план программы «**  Разработка интернет-приложений на платформе C# ASP.Net MVC **»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль / Тема** | **Всего, час** | **Виды учебных занятий** | | | **Формы контроля** |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | Основы работы с ASP.NET MVC | 12 | 2 | 4 | 6 | Отчет по лабораторной работе |
| 1.1 | Концепция паттерна Model-View-Controller | 2 | 0,5 | 0,5 | 1 | Отчет по лабораторной работе |
| 1.2 | Основы работы с ASP.NET MVC. Веб-приложение MVC без аутентификации, структура проекта | 4 | 0,5 | 1,5 | 2 | Отчет по лабораторной работе |
| 1.3 | Синтаксис Razor. Entity Framework | 6 | 1 | 2 | 3 | Отчет по лабораторной работе |
| 2 | Работа с представлениями. HTML-хелперы | 12 | 2 | 4 | 6 | Отчет по лабораторной работе |
| 2.1 | Работа с представлениями | 4 | 1 | 2 | 3 | Отчет по лабораторной работе |
| 2.2 | Стилизация страниц, мастер-страницы. Частичные представления | 4 | 0,5 | 1 | 1 | Отчет по лабораторной работе |
| 2.3 | HTML-хелперы | 4 | 0,5 | 1 | 2 | Отчет по лабораторной работе |
| 3 | Модели и контроллеры | 18 | 2 | 6 | 10 | Отчет по лабораторной работе |
| 3.1 | Модели и контроллеры. Модели вида один-ко-многим | 10 | 1 | 3 | 5 | Отчет по лабораторной работе |
| 3.2 | Методы и представления | 4 | 0,5 | 2 | 3 | Отчет по лабораторной работе |
| 3.3 | Шаблоны формирования | 4 | 0,5 | 1 | 2 | Отчет по лабораторной работе |
| 4 | Пагинация. Фильтрация. Валидация | 12 | 2 | 4 | 6 | Отчет по лабораторной работе |
| 4.1 | Понятие пагинации. Реализация пагинации | 4 | 0,5 | 1 | 1,5 | Отчет по лабораторной работе |
| 4.2 | Фильтрация. Реализация фильтрации | 4 | 1 | 2 | 3 | Отчет по лабораторной работе |
| 4.3 | Валидация. Реализация валидации данных с использованием атрибутов | 4 | 0,5 | 1 | 1,5 | Отчет по лабораторной работе |
| 5 | Веб-приложение MVC с аутентификацией | 16 | 2 | 6 | 8 | Отчет по лабораторной работе |
| 5.1 | Авторизация и аутентификация. Веб-приложение MVC с аутентификацией, структура проекта. | 4 | 0,5 | 2 | 3 | Отчет по лабораторной работе |
| 5.2 | Модели в приложении с аутентификацией. Использование механизма миграций для обновления базы данных проекта | 6 | 1 | 3 | 4 | Отчет по лабораторной работе |
| 5.3 | Распределение ролей пользователей, управление доступом | 6 | 0,5 | 1 | 1 | Отчет по лабораторной работе |
| 6 | Итоговая аттестация | 2 | - | 2 | - | Защита проекта в форме онлайн-собеседования |

**7. Учебная (рабочая) программа повышения квалификации «**  Разработка интернет-приложений на платформе C# ASP.Net MVC **»**

Раздел 1. Основы работы с ASP.NET MVC (12 часов)

Тема 1.1. Концепция паттерна Model-View-Controller (2 часа)

Тема 1.2. Основы работы с ASP.NET MVC. Веб-приложение MVC без аутентификации, структура проекта (4 часа)

Тема 1.3. Синтаксис Razor. Entity Framework (6 часов)

Раздел 2. Работа с представлениями. HTML-хелперы (12 часов)

Тема 2.1. Работа с представлениями (4 часа)

Тема 2.2. Стилизация страниц, мастер-страницы. Частичные представления (4 часа)

Тема 2.3. HTML-хелперы (4 часов)

Раздел 3. Модели и контроллеры (18 часов)

Тема 3.1. Модели и контроллеры. Модели вида один-ко-многим (10 часов)

Тема 3.2. Методы и представления. (4 часа)

Тема 3.3. Шаблоны формирования (4 часа)

Раздел 4. Пагинация. Фильтрация. Валидация (12 часов)

Тема 4.1. Понятие пагинации. Реализация пагинации (4 часа)

Тема 4.2. Фильтрация. Реализация фильтрации (4 часа)

Тема 4.3. Валидация. Реализация валидации данных с использованием атрибутов (4 часа)

Раздел 5. Веб-приложение MVC с аутентификацией (16 часов)

Тема 5.1. Авторизация и аутентификация. Веб-приложение MVC с аутентификацией, структура проекта. (4 часа)

Тема 5.2. Модели в приложении с аутентификацией. Использование механизма миграций для обновления базы данных проекта (6 часов)

Тема 5.3. Распределение ролей пользователей, управление доступом (6 часов)

Итоговая аттестация (2 часа)

**Описание практико-ориентированных заданий и кейсов**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Номер темы/модуля** | **Наименование практического занятия** | **Описание** |
| **1.1** | 1 | Основы работы с ASP.NET MVC | Обучающиеся создают MVC веб-приложение без аутентификации в соответствии с выбранной предметной областью, подключают Entity Framework, создают модели данных, контроллер и представления |
| **1.2** | 2 | Работа с представлениями. HTML-хелперы | Обучающиеся закрепляют работу с представлениями, учатся стилизации страниц, начинают работать с HTML-хелперами в синтаксисе Razor |
| 1.3 | 3 | Модели и контроллеры. | Обучающиеся работают с моделью данных «один-ко-многим», создают в контроллере методы для обработки и управления данными, учатся использовать шаблоны формирования для генерации представлений и других элементов приложения |
| 1.4 | 4 | Пагинация. Фильтрация. Валидация | Обучающиеся реализуют методы контроллера и соответствующее представление для пагинации на странице, реализуют возможность фильтрации данных в представлении, настраивают валидацию вводимых данных |
| 1.5 | 5 | Веб-приложение MVC с аутентификацией | Обучающиеся создают MVC веб-приложение с аутентификацией в соответствии с выбранной предметной областью, учатся выполнять миграцию данных в случае модификации модели данных, определяют роли пользователей созданного приложения, распределяют права доступа в соответствии с назначенной ролью пользователя. |

**8.Оценочные материалы по образовательной программе**

**8.1. Вопросы тестирования по модулям**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ модуля** | **Вопросы входного тестирования** | **Вопросы промежуточного тестирования** | **Вопросы итогового тестирования** |
| **1.1** | Не предусмотрены | Не предусмотрены | Не предусмотрены |

**8.2.**  Отчет по лабораторной работе

Защита проекта в форме онлайн-собеседования

Показатели

1. Полнота выполнения практического задания (в лабораторной работе) и отчета по нему;

2. своевременность выполнения задания;

3. последовательность и рациональность выполнения задания;

4. самостоятельность выполнения. **.**

**8.3.**  Лабораторная работа №3. Модели и контроллеры. Модели вида один-ко-многим

Первым делом подготовьте файл отчета в Microsoft Word (имя файла должно отражать фамилию исполнителя и номер лабораторной работы). На титульном листе и в колонтитуле укажите свою тему и ФИО.

1. Создайте свое MVC веб-приложение без аутентификации.
2. Добавьте в свой проект базу данных, щелкнув правой кнопкой мыши на App\_Data и выбрав в контекстном меню добавление нового элемента. Дайте своей БД осмысленное наименование, связанное с выбранной темой.
3. Используя обозреватель серверов, выберите созданную БД и добавьте в нее 2 таблицы, одна из которых (Employees) будет содержать список сотрудников, а вторая (Products) - список товаров (или услуг), в реализации которых будут участвовать сотрудники. В процессе создания таблиц наблюдайте за изменениями SQL-скрипта в нижней части экрана. Имя таблице можно задать прямо в тексте этого скрипта, указав его вместо Table. Обдумайте, какие поля будут в Вашей БД, какие для них необходимы типы данных, какие поля можно будет иногда оставлять незаполненными (например, отчество). Используйте автоинкрементацию для ключевых полей. В таблице Products предусмотрите поле для связи с таблицей Employees и разрешите его оставлять незаполненным. Например, таблица книг может быть связана с таблицей авторов по AuthorId с помощью SQL-кода:

[AuthorId] INT NULL,

CONSTRAINT [FK\_Books\_Authors] FOREIGN KEY ([AuthorId]) REFERENCES [Authors]([Id]) ON DELETE SET NULL

Последняя часть этого выражения (ON DELETE SET NULL) указывает, что при удалении объекта из таблицы Authors, свойству AuthorId, которое ссылалось на удаленный объект, будет присвоено значение null (чтобы книги не относились к удаленным авторам при их удалении из таблицы авторов).

1. Сделайте и вставьте в отчет скриншоты готовых SQL-скриптов, прокомментируйте их.
2. Внесите данные в таблицы (По 5-10 записей). Проиллюстрируйте это скриншотами в отчете.
3. Теперь задействуйте в своем проекте Entity Framework.
4. Далее добавьте в проект модели сотрудника и товара (услуги). Названия полей созданных ранее таблиц должны совпадать с названиями свойств моделей. В модели товара (услуги) не забудьте, что необходимо поле для связи этой модели с моделью сотрудника. Также добавьте навигационное свойство Employee. Пример класса Book:

public class Book

{

public int Id { get; set; }

public string NameBook { get; set; }

public int YearCreate { get; set; }

public int? AuthorId { get; set; } // "?" учитывает, что AuthorId может быть целочисленным или иметь значение null

public Author Author { get; set; } // Навигационное свойство

}

1. Добавьте класс контекста данных.

1. Откройте web.config, обеспечьте связь с базой данных с помощью строки подключения.
2. Обратитесь к HomeController, подключите свое пространство имен и создайте объект db, через который приложение будет взаимодействовать с базой данных.
3. В методе Index передайте все товары (услуги) в представление в виде списка. Для этого создайте объект products, в который с помощью навигационного свойства Employee получите методом Include перечень товаров (услуг).

Пример метода Index:

public ActionResult Index()

{

var books = db.Books.Include(b => b.Author);// метод Include подгружает связанные данные списка книг из таблицы авторов с помощью лямбда-выражения b => b.Author

return View(books.ToList()); //отправляем список книг в представление

}

1. Пересоздайте представление Index. Для этого надо вызвать контекстное меню, щелкнув правой кнопкой мыши на имени метода Index, выберите опцию добавления представления, в качестве шаблона укажите List, в качестве модели - Product. В полученном представлении лучше заменить Id сотрудника на его личные данные (например, Name).
2. Для добавления новых данных в таблицу Products подготовьте GET и POST версии метода Create и соответствующее представление. Пример метода Create:

[HttpGet]

public ActionResult Create()

{

// Формируем список авторов для передачи в представление

SelectList authors = new SelectList(db.Authors, "Id", "Name");

ViewBag.Authors = authors;

return View();

}

[HttpPost]

public ActionResult Create(Book book)

{

//Добавляем книгу в таблицу

db.Books.Add(book);

db.SaveChanges();

// перенаправляем на главную страницу

return RedirectToAction("Index");

}

Пример представления Create:

@model BooksStore.Models.Book

@{

ViewBag.Title = "Create";

}

<h2>Добавление новой книги</h2>

@using (Html.BeginForm())

{

<fieldset>

<legend>Книга</legend>

<p>

Название <br />

@Html.EditorFor(model => model.NameBook)

</p>

<p>

Год создания <br />

@Html.EditorFor(model => model.YearCreate)

</p>

<p>

Автор <br />

@Html.DropDownListFor(model => model.AuthorId, ViewBag.Authors as SelectList)

</p>

<p>

<input type="submit" value="Добавить книгу" />

</p>

</fieldset>

}

<div>

@Html.ActionLink("К списку книг", "Index")

</div>

1. Аналогично добавьте метод Edit для редактирования данных и соответствующее представление (это представления почти полностью совпадает с представлением Create, только для идентификации записи в таблице необходимо добавить ссылку на Id товара или услуги). Пример метода Edit:

[HttpGet]

public ActionResult Edit(int? id)

{

if (id == null)

{

return HttpNotFound();

}

// Находим в бд книгу

Book book = db.Books.Find(id);

if (book != null)

{

// Создаем список книг для передачи в представление

SelectList books = new SelectList(db.Authors, "Id", "Name", book.AuthorId);

ViewBag.Authors = books;

return View(book);

}

return RedirectToAction("Index");

}

[HttpPost]

public ActionResult Edit(Book book)

{

db.Entry(book).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

Пример представления Edit:

@model BooksStore.Models.Book

@{

ViewBag.Title = "Edit";

}

<h2>Редактирование книги</h2>

@using (Html.BeginForm())

{

<fieldset>

<legend>Книга</legend>

@Html.HiddenFor(model => model.Id)

<p>

Название <br />

@Html.EditorFor(model => model.NameBook)

</p>

<p>

Год создания <br />

@Html.EditorFor(model => model.YearCreate)

</p>

<p>

Автор <br />

@Html.DropDownListFor(model => model.AuthorId, ViewBag.Authors as SelectList)

</p>

<p>

<input type="submit" value="Отредактировать книгу" />

</p>

</fieldset>

}

<div>

@Html.ActionLink("К списку книг", "Index")

</div>

1. Разработайте метод Details и соответсвующее ему представление, выводящее информацию о товаре (услуге).
2. Осталось добавить возможность удаления записей с помощью метода Delete. Пример метода Edit:

[HttpGet]

public ActionResult Delete(int? id)

{

if (id == null)

{

return HttpNotFound();

}

// Находим в бд книгу

Book book = db.Books.Find(id);

if (book != null)

{

db.Books.Remove(book);

db.SaveChanges();

}

return RedirectToAction("Index");

}

1. Сделайте скриншоты, демонстрирующие работу приложения, и вставьте их в файл отчёта, прокомментируйте их. Также в отчете сделайте ссылку на архив своего приложения, который разместите на доступном для Вас git-сервисе. Учтите, что в 4-й лабораторной работе Вам потребуется копия проекта, который Вы разработали в ходе выполнения текущей лабораторной работы.

**.**

**8.4.**  **Обучающие задачи (кейсы)**: Основы работы с ASP.NET MVC

Работа с представлениями. HTML-хелперы

Модели и контроллеры.

Пагинация. Фильтрация. Валидация

Веб-приложение MVC с аутентификацией **.**

**8.5.**  Отчет по лабораторной работе

Защита проекта в форме онлайн-собеседования

Показатели

1. Полнота выполнения практического задания (в лабораторной работе) и отчета по нему;

2. своевременность выполнения задания;

3. последовательность и рациональность выполнения задания;

4. самостоятельность выполнения. **.**

**9.Организационно-педагогические условия реализации программы**

**9.1. Кадровое обеспечение программы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№**  **п/п** | **Фамилия, имя, отчество (при наличии)** | **Место основной работы и должность, ученая степень и ученое звание (при наличии)** | **Ссылки на веб-страницы с портфолио (при наличии)** | **Фото в формате jpeg** | **Отметка о полученном согласии на обработку персональных данных** |
| **1** | Козлов Денис Юрьевич | ФГБОУ ВО «Алтайский государственный университет», заведующий кафедрой информатики, к.ф.-м.н., доцент | https://www.asu.ru/search/persons/1586/ | **Козлов-Денис-Юрьевич** | Да |

**9.2.Учебно-методическое обеспечение и информационное сопровождение**

|  |  |
| --- | --- |
| **Учебно-методические материалы** | |
| Методы, формы и технологии | Методические разработки,  материалы курса, учебная литература |
| Лабораторные работы | Методические разработки к лабораторным работам в курсе на открытом образовательном портале Алтайского государственного университета |
| Лабораторные работы | Список основной литературы   1. Фримен Адам. ASP.NET Core MVC 2 с примерами на C# для профессионалов. - Диалектика-Вильямс, 2019 г. - 1007 с. 2. Троелсен Эндрю, Джепикс Филипп. Язык программирования C# 7 и платформы .NET и .NET Core. - Вильямс, 2018 г. - 1328 с.   Список дополнительной литературы   1. Фримен Адам. ASP.NET MVC 5 с примерами на C# 5.0 для профессионалов», 5-е издание, Вильямс, 2014 г. - 736 с. 2. Прайс Марк Дж. C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов - Питер, 2018 г. - 640 с. |

|  |  |
| --- | --- |
| **Информационное сопровождение** | |
| Электронные  образовательные ресурсы | Электронные  информационные ресурсы |
| Курс на открытом образовательном портале Алтайского государственного университета | https://public.edu.asu.ru/course/view.php?id=879 |
| Начало работы с ASP.NET MVC 5 | URL: <https://docs.microsoft.com/ru-ru/aspnet/mvc/overview/getting-started/introduction/getting-started> |
| Руководство по ASP.NET MVC 5 | <https://metanit.com/sharp/mvc5/> |

**9.3.Материально-технические условия реализации программы**

|  |  |
| --- | --- |
| Вид занятий | Наименование оборудования,  программного обеспечения |
| Лекции | Компьютер с комплектацией:   * + процессор не ниже Core 2 DUO 2.4Hz;   + ОЗУ: не ниже 4GB;   ОС: WINDOWS 7 или выше;  Discord / Zoom, Google Chrome / Yandex Browser |
| Лабораторные работы | Компьютер с комплектацией:   * + процессор не ниже Core 2 DUO 2.4Hz;   + ОЗУ: не ниже 4GB;   + ОС: WINDOWS 7 или выше;   + Браузер: Google Chrome, Yandex.Browser;   + Visual Studio 2019 с установленными компонентами для разработки веб-приложений (Asp.Net MVC)   + Microsoft SQL Server (не ниже 2014)   + доступ к образовательному пространству, расположенному по адресу: <https://public.edu.asu.ru/>, где расположены материалы для самостоятельного изучения и выполнения. |

**III.Паспорт компетенций (Приложение 2)**

Описание перечня профессиональных компетенций, качественное изменение которых осуществляется в результате обучения.

Планируемые результаты обучения должны быть определены в виде знаний, умений, навыков и (или) опыта деятельности, характеризующих формирование/развитие компетенции(-й) в области цифровой экономики и представлены в виде Паспорта компетенций в машиночитаемом текстовом формате. Структура паспорта представлена в приложении.

ПАСПОРТ КОМПЕТЕНЦИИ

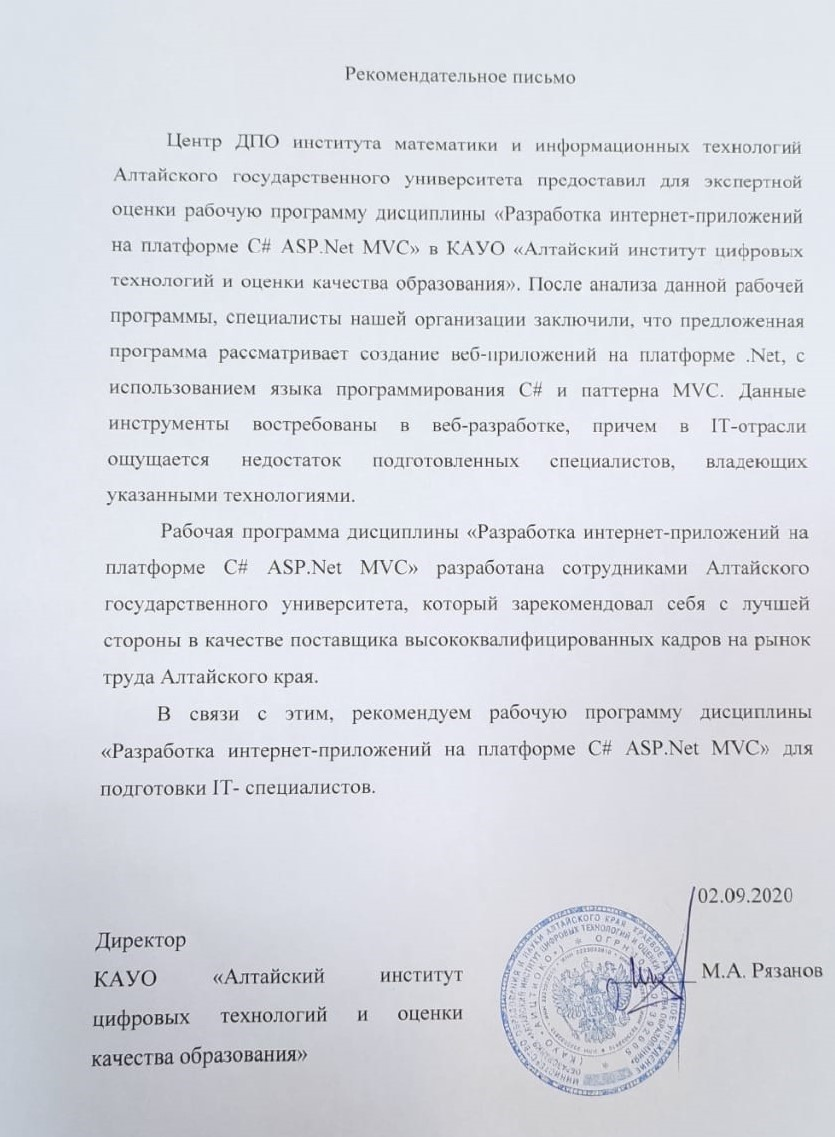
Разработка интернет-приложений на платформе C# ASP.Net MVC

Центр ДПО ИМИТ АлтГУ

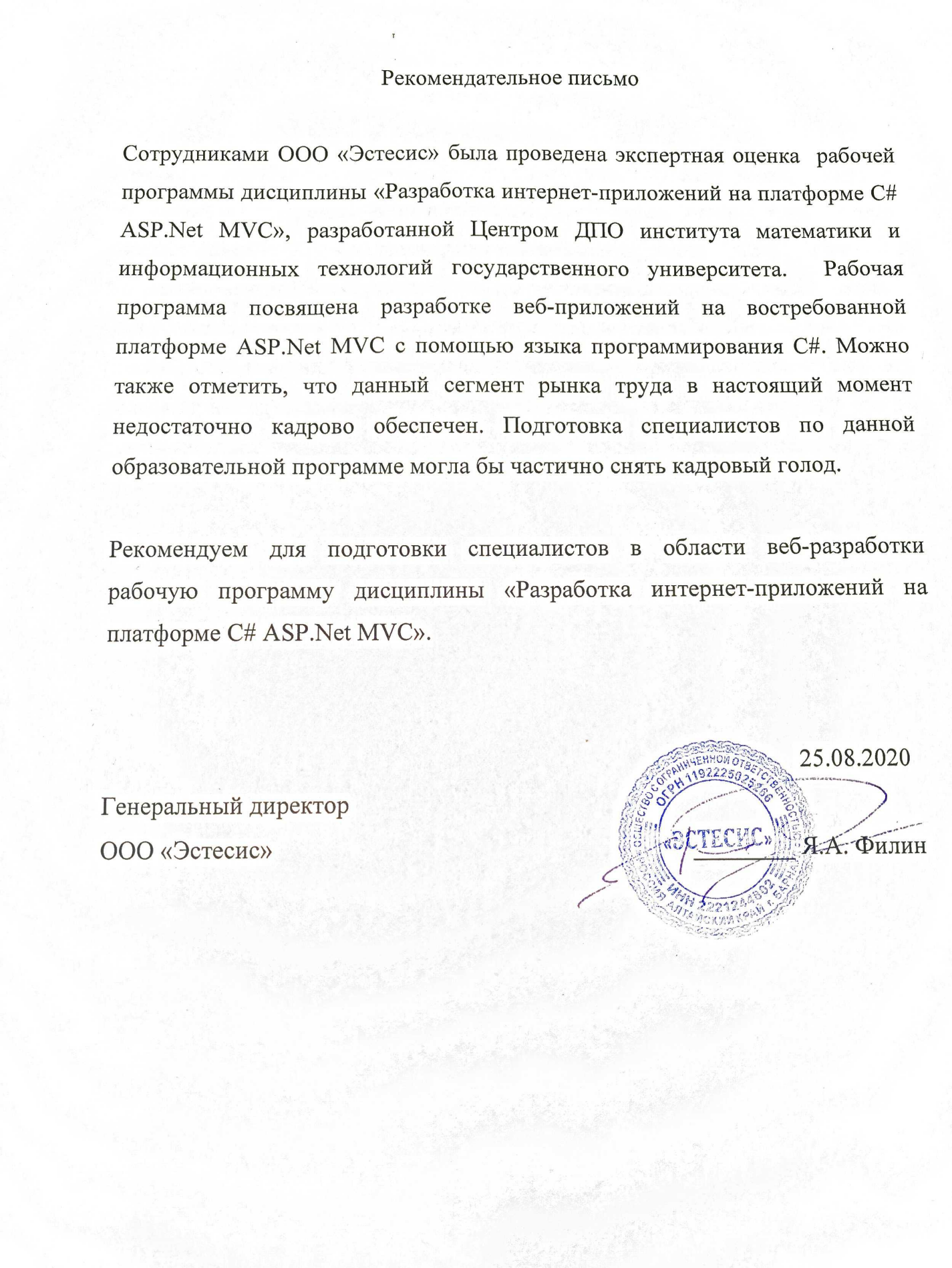
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Наименование компетенции | | Способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение - интернет-приложения на платформе .Net | |
|  | Указание типа компетенции | общекультурная/ универсальная | Профессионально-специализированная | |
| общепрофессиональная |
| Профессионально-специализированная |
|  | Определение, содержание и основные сущностные характеристики компетенции | | В результате освоения данного курса обучающиеся должны знать:  структуру и принципы использования технологии ASP.Net для создания Интернет-приложений.  Обучающиеся должны владеть:  концепцией архитектурного паттерна MVC технологиями,  уметь:  разрабатывать Интернет приложения на языке C#, использовать MS'Visual Studio, Entity Framework, создавать защищенные Интернет-приложения с разграничением доступа в соответствии с политикой ролей. | |
|  | Дескриптор знаний, умений и навыков по уровням | | Уровень сформированности компетенции обучающегося | Индикаторы |
| Начальный | Знает:   |  | | --- | | * концепцию паттерна MVC для веб -приложений, принципы ООП, основы C# * структуру проекта приложения ASP.Net MVC5 |   Умеет:   |  | | --- | | * создавать модели, контроллеры и представления в проекте приложения ASP.Net MVC5 |   Владеет:   |  | | --- | | * навыками использования среды разработки MS Visual Studio. | | * навыками программирования на C#. * навыками использования Entity Framework | |
| Базовый | Знает:   |  | | --- | | * синтаксис Razor * способы настройки интерфейса приложения с помощью мастер-страниц, стилевых файлов и частичных представлений |   Умеет:   |  | | --- | | * создавать веб-приложения MVC5 без аутентификации с аутентификации в среде разработки MS Visual Studio . |   Владеет:   |  | | --- | | * создания моделей, контроллеров и представлений в проекте приложения ASP.Net MVC5 | |  | |
| Продвинутый (Владеет сложными навыками, способен активно влиять на происходящее, проявлять соответствующие навыки в ситуациях повышенной сложности.) | Знает:   |  | | --- | | * способы настройки валидации данных с помощью атрибутов |   Умеет:   |  | | --- | | * разграничивать права пользователей с соответствии с заданными ролями * использовать пагинацию |   Владеет:   |  | | --- | | * навыками осуществления фильтрации | |  | |
|  | Характеристика взаимосвязи данной компетенции с другими компетенциями / необходимость владения другими компетенциями для формирования данной компетенции | | Данная компетенция необходима для профессиональной компетенции способность программировать приложения и создавать программные прототипы решения прикладных задач в области веб-разработки. | |
|  | Средства и технологии оценки | | Отчет по лабораторной работе  Защита проекта в форме онлайн-собеседования  Показатели  1. Полнота выполнения практического задания (в лабораторной работе) и отчета по нему;  2. своевременность выполнения задания;  3. последовательность и рациональность выполнения задания;  4. самостоятельность выполнения. | |

**VI.Иная информация о качестве и востребованности образовательной программы** (результаты профессионально-общественной аккредитации образовательной программы, включение в системы рейтингования, призовые места по результатам проведения конкурсов образовательных программ и др.) (при наличии)

**V.Рекомендаций к программе от работодателей**: наличие не менее двух писем и/или подтверждения на цифровой платформе Государственной системы предоставления ПЦС от работодателей о рекомендации образовательной программы для реализации в рамках Государственной системы предоставления ПЦС на формирование у трудоспособного населения компетенций цифровой экономики с указанием востребованности результатов освоения программы в сфере деятельности соответствующих компаний и готовности к рассмотрению заявок наиболее успешно освоивших образовательную программу граждан на прохождение стажировки и (или) собеседования на предмет трудоустройства путем проставления отметки в профиле программы

РекомендательноеПисьмоАИЦТиОКО.pdf

РекомендательноеПисьмоЭстесис.pdf



**VI.Указание на возможные сценарии профессиональной траектории граждан** по итогам освоения образовательной программы (в соответствии с приложением)

Возможные сценарии профессиональной траектории граждан по итогам освоения образовательной программы

1. Освоение образовательной программы работающими гражданами на должности с требованиями навыков разработки на платформе C# ASP.Net MVC -> использование навыков в профессиональной деятельности.
2. Освоение образовательной программы работающими гражданами -> переход на должность с требованиями навыков разработки на платформе C# ASP.Net MVC -> использование навыков в профессиональной деятельности.
3. Освоение образовательной программы неработающими гражданами -> устройство на работу с требованиями навыков разработки на платформе C# ASP.Net MVC -> использование навыков в профессиональной деятельности.

**VII.Дополнительная информация**

**VIII.Приложенные Скан-копии**

Утвержденной рабочей программа (подпись, печать, в формате pdf)