

Банк заданий по базовой части вступительного испытания в магистратуру

Задание 1 (10 баллов)

1.1. Вопрос:

Известно, что объем y спроса на некоторый товар и его цена x связаны уравнением парной линейной регрессии. Оценка коэффициента уравнения равна -0.5 , оценка сдвига уравнения равна 200 . Дайте прогноз y при $x=20$.

Решение:

Уравнение парной линейной регрессии имеет вид: $y=mx+b+\varepsilon$, где x – объясняющая переменная, y – объясняемая переменная, m и b – параметры уравнения, ε – случайная величина, возмущение.

Общая формула для всех задач: Прогноз $y = (\text{Коэффициент наклона}) * (\text{Значение } x) + (\text{Свободный член} / \text{Константа} / \text{Сдвиг})$

Прогнозируемое значение y определяется как оценка y^{\wedge} , получаемая по формуле: $y^{\wedge}=m^{\wedge}x+b^{\wedge}$, где m^{\wedge} и b^{\wedge} – оценка параметров m и b .

Подставляя в формулу значения $m^{\wedge}=-0,5$, $b^{\wedge}=200$, $x=20$, получим $y^{\wedge}=190$

$Y^{\wedge}=-0.5*20+200=-10+200=190$.

Ответ: 190.

1.2. Вопрос:

Зависимость расхода топлива автомобиля y (л/100км) от его мощности двигателя x (л.с.) описывается уравнением регрессии с угловым коэффициентом 0.04 и сдвигом 5 . Дайте прогноз расхода топлива для автомобиля мощностью 150 л.с.

1.3. Вопрос:

Экономист построил линейную модель, связывающую уровень безработицы в регионе x (%) с уровнем преступности y (преступлений на 1000 чел.). Оценка коэффициента уравнения равна 2.1 , оценка сдвига – 1.5 . Найдите прогнозируемый уровень преступности при уровне безработицы 7.5% .

1.4. Вопрос:

Уравнение парной линейной регрессии, связывающее среднюю температуру в июле x ($^{\circ}\text{C}$) и урожайность пшеницы y (ц/га), имеет оценку коэффициента наклона 1.8 и оценку сдвига -10 . Дайте прогноз урожайности для региона, где средняя июльская температура составляет 22°C .

1.5. Вопрос:

Линейная регрессионная модель зависимости количества проданных единиц товара y от его цены x (у.е.) построена с оценками: коэффициент при цене равен -12 , а свободный член равен 500 . Каков прогноз продаж, если цена установлена на уровне 30 у.е.?

Задание 2 (10 баллов)

2.1. Вопрос:

Можно ли считать уравнение линейной регрессии значимым, если коэффициент детерминации равен 0.75?

Решение:

Уравнение линейной регрессии можно считать значимым, если коэффициент детерминации = 0,75, т.к. коэффициент детерминации показывает, какая часть изменения зависимой переменной объясняется изменением объясняющей переменной. $0 \leq R^2 \leq 1$. Чем ближе R^2 к 1, тем лучше регрессия аппроксимирует наблюдения.

$R^2 \geq 0,5$ ($r \geq 0.7$) - регрессионная модель значима, так как ближе к 1.

Ответ: Да.

2.2. Вопрос:

Определить абсолютное значение коэффициента корреляции между переменными X и Y, при $R^2 = 0,15$. Является ли связь между этими переменными тесной?

2.3. Вопрос:

Определить абсолютное значение коэффициента корреляции между переменными X и Y, при $R^2 = 0,34$. Является ли связь между этими переменными тесной?

2.4. Вопрос:

Определить абсолютное значение коэффициента корреляции между переменными X и Y, при $R^2 = 0,84$. Является ли связь между этими переменными тесной?

2.5. Вопрос:

Определить абсолютное значение коэффициента корреляции между переменными X и Y, при $R^2 = 0,56$. Является ли связь между этими переменными тесной?

Задание 3 (10 баллов)

3.1. Вопрос:

В условиях классической нормальной линейной регрессионной модели коэффициент детерминации $R^2=0.9$, полная сумма квадратов равна 30. Чему равна регрессионная сумма квадратов?

Решение:

$R^2=QR/Q$, где R-коэффициент детерминации; QR-регрессионная сумма; Q-полная сумма квадратов. $QR= R^2*Q$.

$$QR=0.9*30=27.$$

Ответ: 27.

3.2. Вопрос:

В условиях классической нормальной линейной регрессионной модели коэффициент детерминации $R^2 = 0.12$, полная сумма квадратов равна 500. Чему равна остаточная сумма квадратов?

3.3. Вопрос:

В условиях классической нормальной линейной регрессионной модели коэффициент детерминации $R^2 = 0.0$, полная сумма квадратов равна 90. Чему равна регрессионная сумма квадратов?

3.4. Вопрос:

В условиях классической нормальной линейной регрессионной модели коэффициент детерминации $R^2 = 0.67$, полная сумма квадратов равна 60. Чему равна остаточная сумма квадратов?

3.5. Вопрос:

В условиях классической нормальной линейной регрессионной модели коэффициент детерминации $R^2 = 0.3$, полная сумма квадратов равна 200. Чему равна регрессионная сумма квадратов?

Задание 4 (10 баллов)

4.1.Вопрос:

Что такое временной ряд в эконометрике?

Ответ:

Временной ряд – это совокупность значений, какого – либо показателя за несколько последовательных моментов или периодов времени. Каждое значение (уровень) временного ряда формируется под воздействием большого числа факторов, которые можно факторы:

- формирующие тенденцию ряда;
- факторы, формирующие циклические колебания ряда;
- случайные факторы.

Основная задача эконометрического исследования отдельного временного ряда – выявление количественного выражения каждой из компонент и использование полученной информации для прогноза будущих значений ряда или построение модели взаимосвязи двух или более временных рядов.

4.2.Вопрос:

Что такое интервенция временного ряда?

4.3.Вопрос:

Что такое панельные данные и в чём их основное преимущество перед чистыми временными рядами или пространственными срезами?

4.4.Вопрос:

Что такое возмущение в эконометрической модели?

Банк заданий по специальной части вступительного испытания в магистратуру

Задание 5 (15 баллов)

5.1. Вопрос:

Принципы SOLID.

Ответ:

SOLID — это набор пяти принципов проектирования.

1. Принцип единственной ответственности (Single Responsibility Principle, SRP):

Класс должен иметь только одну причину для изменения, то есть выполнять лишь одну задачу.

Пример нарушения:

```
public class Order
{
    public void CalculateTotal() { /* Логика расчёта суммы заказа */ }
    public void PrintInvoice() { /* Логика печати счёта */ }
}
```

Правильный подход – разделить класс Order на два класса.

2. Принцип открытости/закрытости (Open/Closed Principle, OCP):

Классы должны быть открыты для расширения, но закрыты для модификации.

Пример нарушения:

```
public double CalculateArea(object shape)
{
    if (shape is Rectangle) { /* Расчёт площади прямоугольника */ }
    else if (shape is Circle) { /* Расчёт площади круга */ }
    // При добавлении новой фигуры придётся изменять этот метод
}
```

Правильный подход – использовать интерфейс IShape с методом CalculateArea и наследование.

3. Принцип подстановки Барбары Лисков (Liskov Substitution Principle, LSP):

Подклассы должны заменять свои базовые классы, не ломая логику программы.

Пример нарушения:

```
public class Bird
{
    public virtual void Fly() { /* ... */ }
}
public class Penguin : Bird
{
    public override void Fly() => throw new NotSupportedException("Пингвины не летают!");
}
```

Правильный подход – исключить наследование Penguin от Bird. Ввести более специализированные интерфейсы.

4. Принцип разделения интерфейсов (Interface Segregation Principle, ISP):

Программные сущности не должны зависеть от методов, которые они не используют. Интерфейсы не должны включать в себя слишком много методов.

Пример нарушения:

```
public interface IWorker
{
    void Work();
    void Eat();
}

public class Robot : IWorker
{
    public void Work() { /* ... */ }
    public void Eat() => throw new NotSupportedException("Роботы не едят!");
}
```

Правильный подход – разделить интерфейс IWorker. Ввести более специализированные интерфейсы.

5. Принцип инверсии зависимостей (Dependency Inversion Principle, DIP):

Модули верхнего уровня не должны зависеть от модулей нижнего уровня. Оба должны зависеть от абстракций.

Пример нарушения:

```
public class OrderService
{
    private readonly SqlDatabase _database; // Жёсткая зависимость от конкретной БД
}
```

Правильный подход – реализовать интерфейс IDatabase и использовать его наследника SqlDatabase.

Применение принципов SOLID позволяет сделать код более гибким (легко добавлять новую функциональность), поддерживаемым (меньше связанности, проще тестировать), масштабируемым (удобно расширять без переписывания).

5.2. Вопрос:

Задачи анализа данных Data Mining.

Ответ:

Data Mining (интеллектуальный анализ данных) – это процесс обнаружения скрытых закономерностей, аномалий и полезных знаний в больших массивах данных с использованием методов машинного обучения, статистики и алгоритмов.

Основные задачи Data Mining можно разделить на описательные (анализ текущих данных) и предсказательные (прогнозирование будущих событий).

Описательные задачи:

1. Кластеризация (Clustering) – разделение данных на группы (кластеры) по схожести без заранее известных меток.

Применение:

- Сегментация клиентов (например, для персонализации рекламы).
- Группировка документов по темам.

2. Ассоциативные правила (Association Rules) – поиск взаимосвязей между событиями или объектами в транзакционных данных.

Применение:

Анализ покупок в супермаркете (правило "Если купили угли для шашлыка, то часто берут смесь для розжига").

Рекомендательные системы (Amazon, Netflix).

3. Анализ аномалий (Anomaly Detection) – обнаружение нестандартных данных (выбросов).

Применение:

Выявление мошеннических транзакций.

Обнаружение сбоев в промышленном оборудовании.

Предсказательные задачи

1. Классификация (Classification) – отнесение объектов к заранее определённым классам.

Применение:

Определение спама в email.

Диагностика заболеваний по симптомам.

2. Регрессия (Regression) – прогнозирование числовых значений на основе исторических данных.

Применение:

Предсказание цен на недвижимость.

Прогнозирование спроса на товары.

3. Прогнозирование временных рядов (Time Series Forecasting) – предсказание будущих значений на основе временных данных.

Применение:

Прогноз курса акций.

Планирование нагрузки на серверы.

Прочие задачи:

1. Снижение размерности (Dimensionality Reduction) – упрощение данных без потери ключевой информации.

Применение:

Визуализация многомерных данных.

Ускорение обучения моделей.

2. Анализ текстов (Text Mining) – извлечение смысла из неструктурированных текстов.

Применение:

Анализ отзывов клиентов.

Классификация новостей.

Data Mining решает широкий спектр задач: от поиска скрытых закономерностей до точных прогнозов. Методы Data Mining применяются в финансах, медицине, ритейле, IT и других областях.

5.3.Вопрос:

Машинный язык. Ассемблер. Языки программирования высокого уровня.

5.4.Вопрос:

Структурное программирование. Принципы структурного подхода к разработке ПО.

5.5.Вопрос:

Задача кластеризации. Применяемые методы.

5.6.Вопрос:

Деревья решений. Процесс конструирования.

Задание 6 (15 баллов)

6.1. Вопрос:

Понятие БД. Основные свойства современных СУБД.

Ответ:

База данных (БД) – это организованная, структурированная совокупность данных, хранящихся в электронном виде и управляемых с помощью специализированного программного обеспечения – системы управления базами данных (СУБД). СУБД – это совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

Ключевые характеристики БД (обеспечиваются СУБД):

- Структурированность – данные организованы по определённым правилам (таблицы, документы, графы и т. д.).
- Постоянность – данные сохраняются после завершения работы программы.
- Доступность – возможность извлечения и модификации данных через интерфейсы (SQL, API и др.).
- Целостность – данные соответствуют заданным правилам и ограничениям.

Современные системы управления базами данных (СУБД) обладают следующими ключевыми свойствами:

- Надёжность и отказоустойчивость

Транзакции (ACID) – гарантия атомарности, согласованности, изолированности и долговечности операций.

Репликация – автоматическое копирование данных на несколько серверов для защиты от сбоев (например, в PostgreSQL можно настроить репликацию master-slave).

Резервное копирование – возможность восстановления данных после аварии.

- Масштабируемость

Горизонтальное масштабирование – распределение нагрузки между несколькими серверами (например, в Postgres).

Вертикальное масштабирование – увеличение мощности одного сервера (например, добавление RAM в SQL-сервер).

- Производительность

Индексация – ускорение поиска данных (B-деревья, хеш-индексы).

Кэширование – хранение часто используемых данных в памяти (например, Redis).

Оптимизация запросов – автоматический выбор эффективного плана выполнения (например, в Postgres).

- Безопасность

Аутентификация и авторизация – контроль доступа пользователей (роли, права).

Шифрование данных – защита информации при хранении и передаче.

Аудит – журналирование действий пользователей.

- Поддержка различных моделей данных

Реляционные (SQL) – таблицы с жёсткой схемой (PostgreSQL, MySQL).

NoSQL – документы (MongoDB), ключ-значение (Redis), графы (Neo4j).

Гибридные — поддержка SQL и NoSQL (например, PostgreSQL с JSONB).

- Распределённость

Шардирование – разделение данных между серверами для балансировки нагрузки.

Поддержка интеграций с облачными решениями.

- **Расширяемость**

Поддержка пользовательских функций – возможность добавлять свои процедуры на различных языках (например, T-SQL, PL/pgSQL).

Плагины и расширения — дополнительные модули для специфичных задач (например, в PostgreSQL).

6.2. Вопрос:

Уровни абстракции БД. Реляционная СУБД.

6.3. Вопрос:

Реляционные СУБД. Понятия домен, кортеж, отношения.

6.4. Вопрос:

Реляционные и нереляционные БД.

Задание 7 (15 баллов)

7.1. Вопрос:

Язык UML. Диаграмма состояний. Пример для предметной области «Магазин».

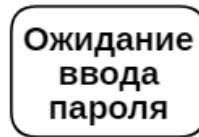
Ответ:

UML – это стандартизированный графический язык моделирования, используемый для визуализации, спецификации, конструирования и документирования систем.

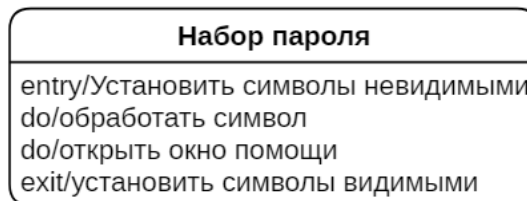
Язык включает более десяти видов диаграмм, одной из которых является диаграмма состояний (диаграмма конечного автомата). Диаграмма состояний (state diagram) показывает автомат, сосредотачивая внимание на потоке управления от одного состояния к другому. Изображается в виде графа с вершинами и дугами (ребрами). Автомат (state machine) – это описание последовательности состояний, через которые проходит объект на протяжении жизненного цикла, реагируя на события, а также описание реакции на эти события.

Основные элементы диаграммы состояний:

Состояния – стабильные условия, в которых может находиться объект. Это ситуация в жизненном цикле объекта, на протяжении которой он удовлетворяет некоторому условию, выполняет некоторую деятельность или ожидает некоторого события. Состояние (простой) изображается прямоугольником с закругленными углами с именем состояния, изображенным внутри этого прямоугольника:



Возможен вариант изображения с разделением на несколько секций. Секциями простого состояния являются: секция имени, секция внутренней деятельности, секция внутренних переходов. Пример изображения:



Деятельность (activity) специфицирует работу, происходящую внутри автомата. Действия – операции, выполняемые при переходе или в состоянии. Это примитивное выполняемое вычисление, приводящее к смене состояния модели или возврату значения.

Начальное псевдосостояние представляет собой частный случай состояния, которое не содержит никаких внутренних действий. Изображается как чёрный круг:



Финальное состояние – специальный вид состояния, предназначенного для моделирования завершения конечного автомата или региона, в котором оно содержится. Изображается как чёрный круг в белой рамке:

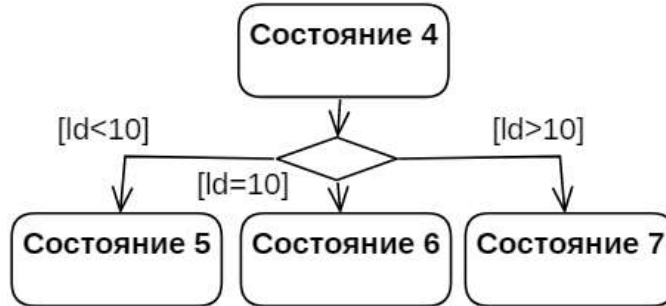


События – триггеры, вызывающие переходы. Это спецификация существенного факта, который происходит во времени и пространстве. В контексте автомата событие – это воздействие, которое вызывает переход между состояниями.

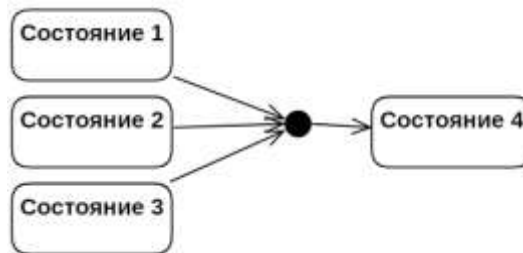
Переходы – изменения состояний под действием событий. Это связь между двумя состояниями, показывающая, что объект, находящийся в первом состоянии, должен выполнить некоторые действия и перейти во второе, как только произойдет определенное событие и будут выполнены определенные условия. Переходы отображаются сплошной линией со стрелкой:



Псевдосостояние выбора предназначено для моделирования нескольких альтернативных ветвей при реализации поведения конечного автомата. Псевдосостояние выбора также называют вершиной или узлом выбора, или просто – выбором. Узел выбора изображается в форме символа ромба. Каждый выходящий из узла выбора переход должен иметь обязательное сторожевое условие, которое представляет собой логическое условие, записанное в квадратных скобках. Пример:

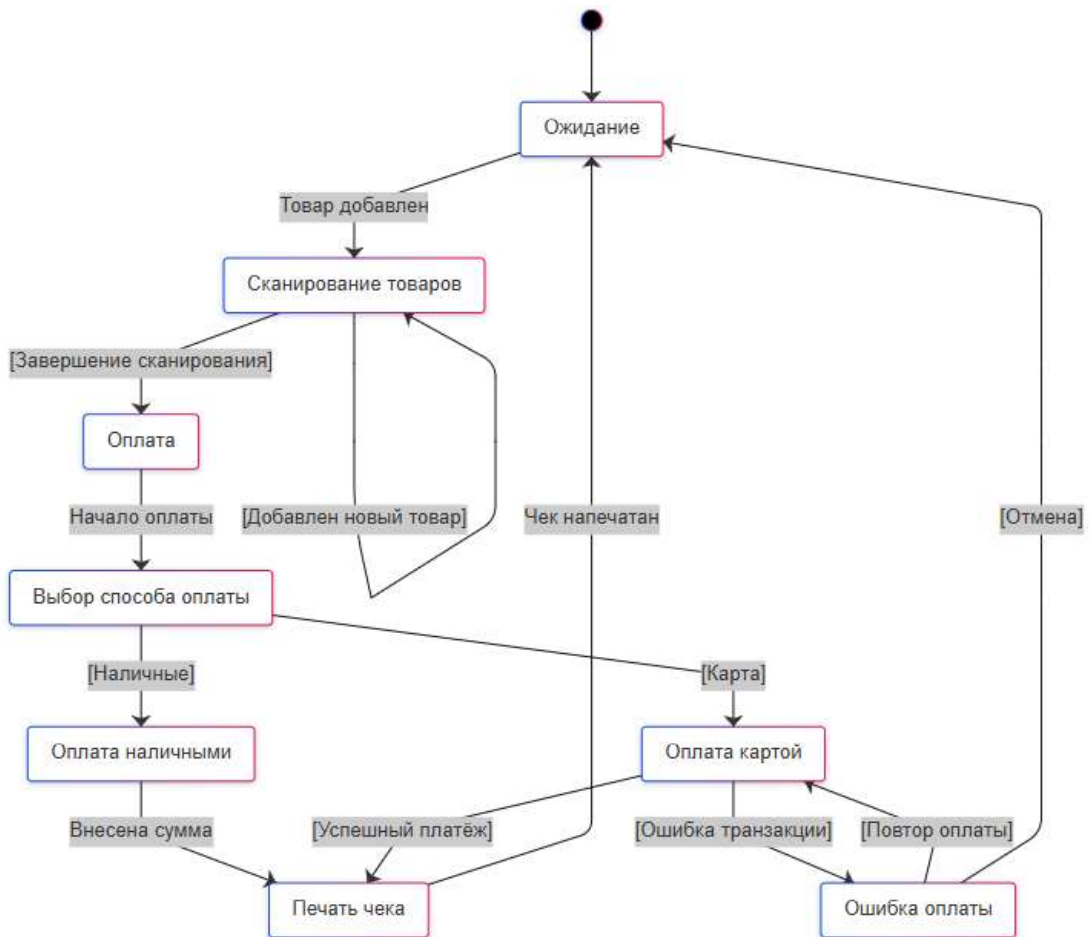


Псевдосостояние соединения является вершиной, которая используется для соединения вместе нескольких переходов. Графически изображается следующим образом:



UML позволяет использовать и более сложные нотации на диаграмме состояний. Например, композитные состояния, в которых присутствуют ортогональные (параллельные) регионы; вершины слияния для описания переходов связанных с ортогональными регионами; точки входа и выхода «из» и «в» композитное состояние.

Пример диаграммы состояний для предметной области «Магазин». Диаграмма охватывает основной цикл работы кассового аппарата: от сканирования товаров до печати чека.



7.2.Вопрос:

Язык UML. Диаграмма классов. Пример для предметной области «Университет».

7.3.Вопрос:

Язык UML. Диаграмма последовательности. Пример для предметной области «Библиотека».

7.4.Вопрос:

Язык UML. Диаграмма вариантов использования. Пример для предметной области «Аквапарк».

7.5.Вопрос:

Язык UML. Диаграмма деятельности. Пример для предметной области «Автовокзал».

Задание 8 (15 баллов)

8.1. Вопрос:

Разработайте модуль программного приложения, позволяющий передавать данные для авторизации пользователя на сервер. Фрагмент схемы БД предложите самостоятельно. Используйте любую известную Вам технологию разработки ПО.

Ответ:

Для разработки используем следующие технологии: платформа ASP.NET Core, ORM библиотека Entity Framework Core, СУБД PostgreSQL.

Таблица Users для хранения данных пользователей:

```
CREATE TABLE Users (  
    Id SERIAL PRIMARY KEY,  
    Username VARCHAR(50) UNIQUE NOT NULL,  
    Email VARCHAR(100) UNIQUE NOT NULL,  
    PasswordHash VARCHAR(255) NOT NULL,  
    Salt VARCHAR(100) NOT NULL  
);
```

Модель пользователя в Entity Framework:

```
public class User  
{  
    public int Id { get; set; }  
    public string Username { get; set; }  
    public string PasswordHash { get; set; }  
    public string Salt { get; set; }  
}
```

Контекст Entity Framework для взаимодействия с БД

```
public class AppDbContext : DbContext  
{  
    ...  
    public DbSet<User> Users { get; set; }  
}
```

// Сервис авторизации

```
public class AuthService  
{  
    private readonly AppDbContext _db;  
    private readonly IConfiguration _config;  
  
    public AuthService(AppDbContext db, IConfiguration config)  
    {  
        _db = db;  
        _config = config;  
    }  
  
    // Попытка входа в систему  
    public async Task<string?> Login(string username, string password)  
    {  
        var user = await _db.Users.FirstOrDefaultAsync(u => u.Username == username);  
        if (user == null)  
            return null;  
    }  
}
```

```

        var inputHash = HashPassword(password, user.Salt);
        if (inputHash != user.PasswordHash)
            return null;
        return GenerateJwtToken(user);
    }

    // Методы HashPassword и GenerateJwtToken опущены для упрощения
}

```

Контроллер для обработки запроса пользователя.

```

public class AuthController : ControllerBase
{
    private readonly AuthService _authService;

    public AuthController(AuthService authService)
    {
        _authService = authService;
    }

    [HttpGet]
    public IActionResult Login()
    {
        return View();
    }

    [HttpPost]
    public async Task<IActionResult> Login(string username, string password)
    {
        var token = await _authService.Login(username, password);
        if(token == null)
            return Unauthorized();

        Response.Cookies.Append("jwt", token);
        // Перенаправляем на главную страницу приложения
        return Redirect("/");
    }
}

```

Форма входа:

```

<form method="post" action="/Auth/Login">
    <div>
        <label>Логин:</label>
        <input type="text" name="username" required />
    </div>
    <div>
        <label>Пароль:</label>
        <input type="password" name="password" required />
    </div>
    <button type="submit">Войти</button>
</form>

```

Дополнительно требуется описать параметры запуска веб-приложения, добавить конфигурацию доступа к БД.

Краткое описание работы:

Пользователь заходит на /Auth/Login, заполняет HTML-форму с логином и паролем.

После отправки формы на сервере определяется пользователь; хешируется переданный пароль с солью, которая была ранее сохранена в БД; полученный хеш проверяется на соответствие хешу из БД. Если хеши совпадают, то генерируется JWT-токен и сохраняется в куки. Если нет – возвращается ошибка.

8.2.Вопрос:

Разработайте модуль программного приложения, позволяющий передавать данные на сервер для добавления товара в корзину в интернет-магазине. Фрагмент схемы БД предложите самостоятельно. Используйте любую известную Вам технологию разработки ПО.

8.3.Вопрос:

Разработайте модуль программного приложения, позволяющий передавать данные на сервер для добавления отзыва о товаре интернет-магазине. Фрагмент схемы БД предложите самостоятельно. Используйте любую известную Вам технологию разработки ПО.

8.4.Вопрос:

Разработайте модуль программного приложения, позволяющий передавать данные на сервер для добавления промокода к заказу в интернет-магазине. Фрагмент схемы БД предложите самостоятельно. Используйте любую известную Вам технологию разработки ПО.