

## Тема 1 – Приложение для учета задач

### Обязательные требования:

1. Разработать базу данных с использованием PostgreSQL, включающую связи:
  - 1-1 (например, пользователь - профиль пользователя)
  - 1-m (например, пользователь - задачи)
  - m-m (например, метки - задачи)
2. Реализовать функционал для:
  - Создания, редактирования, удаления и чтения одной записи задачи.
  - Чтения всех записей задач.
3. Реализовать возможность загрузки файлов к задачам.
4. Реализовать систему регистрации и аутентификации пользователей (логин).
5. Реализовать функционал для смены пароля пользователем.
6. Реализовать функционал для удаления пользователя.

### Необязательные требования:

1. Реализовать систему прав доступа к различным страницам (например, администратор, пользователь).
2. Реализовать возможность аутентификации через Google.
3. Добавить стилизацию с использованием CSS (Bootstrap) и добавить интерактивности с помощью JavaScript.
4. Интегрировать приложение с использованием сторонних API (например, Google Calendar API для синхронизации задач).
5. Добавить функционал отправки уведомлений (например, по электронной почте) о важных событиях или сроках выполнения задач.

### Регистрация:

1. **Страница регистрации:** Пользователь переходит на страницу регистрации, где ему предоставляется форма для ввода информации.
2. **Заполнение формы:** Пользователь заполняет форму, вводя свои данные, такие как имя пользователя, электронная почта и пароль. Дополнительные поля могут включать имя, фамилию и другие необходимые данные.
3. **Проверка данных:** После заполнения формы приложение проверяет введенные данные на корректность и уникальность (например, проверка на уникальность имени пользователя и электронной почты).
4. **Создание учетной записи:** Если данные прошли проверку успешно, приложение создает новую учетную запись пользователя в базе данных.
5. **Успешная регистрация:** Пользователю выводится сообщение об успешной регистрации и, возможно, предлагается автоматический вход в систему.

### Авторизация:

1. **Страница входа:** Пользователь переходит на страницу входа (логина), где ему предоставляется форма для ввода учетных данных.
2. **Ввод данных:** Пользователь вводит свои учетные данные, такие как имя пользователя (или электронная почта) и пароль.
3. **Проверка учетных данных:** Приложение проверяет введенные учетные данные на соответствие данным, хранящимся в базе данных.
4. **Успешная авторизация:** Если введенные данные верны и учетная запись активна, пользователь успешно аутентифицируется и ему предоставляется доступ к защищенным страницам приложения.

5. **Сессия и токены:** После успешной авторизации приложение создает сессию пользователя или выдает ему токен доступа, который используется для последующих запросов к защищенным ресурсам.
6. **Неудачная авторизация:** Если введенные учетные данные неверны или учетная запись не активна, пользователю выводится сообщение об ошибке, и ему предлагается повторить попытку входа или восстановить пароль.

#### **Добавление новой задачи:**

1. **Страница добавления задачи:** Пользователь переходит на страницу, где предоставляется форма для добавления новой задачи.
2. **Заполнение формы:** Пользователь заполняет форму, указывая заголовок задачи, описание, срок выполнения, приоритет и другие необходимые детали.
3. **Сохранение задачи:** После заполнения формы пользователь нажимает кнопку "Добавить" или подтверждает добавление задачи. Введенная информация сохраняется в базе данных.
4. **Уведомление о добавлении:** Пользователю выводится сообщение об успешном добавлении задачи. Возможно, также предоставляется ссылка на страницу со списком всех задач.

#### **Завершение задачи:**

1. **Страница списка задач:** Пользователь переходит на страницу, где отображается список его задач.
2. **Выбор задачи:** Пользователь выбирает задачу, которую он хочет завершить, либо переходит на страницу с конкретной задачей.
3. **Отметка задачи как завершенной:** Пользователь нажимает на соответствующую кнопку или выполняет другое действие для отметки задачи как завершенной.
4. **Обновление статуса задачи:** Приложение обновляет статус задачи в базе данных, отмечая ее как завершенную.
5. **Уведомление об завершении:** Пользователю выводится сообщение о успешном завершении задачи. Возможно, также предоставляется опция перехода к списку задач или странице с завершенными задачами.
6. **Архивация задачи (по желанию):** После завершения задачи, она может быть автоматически перемещена в архив или помечена специальным образом для удобства пользователя.

#### **Просмотр списка задач:**

1. **Авторизация:** Пользователь авторизуется в системе, вводя свои учетные данные на странице входа.
2. **Переход на страницу списка задач:** После успешной авторизации пользователь переходит на страницу, где отображается список всех его задач.
3. **Отображение списка задач:** На странице отображается список задач с информацией о каждой задаче, такой как заголовок, срок выполнения, приоритет и статус.
4. **Доступ к задачам:** Пользователь может просматривать все задачи в списке, а также применять фильтры или сортировку, если такая функциональность предусмотрена.

#### **Просмотр конкретной задачи:**

1. **Выбор задачи из списка:** Пользователь выбирает конкретную задачу из списка задач, кликнув по ней или перейдя на страницу с подробным описанием задачи.
2. **Отображение информации о задаче:** На странице с подробной информацией о задаче отображается ее заголовок, описание, срок выполнения, приоритет, статус и другие детали.

3. **Возможность редактирования:** Пользователь может редактировать задачу, если предусмотрена такая функциональность, например, изменить срок выполнения или приоритет.

#### **Добавление файла к задаче:**

1. **Переход на страницу задачи:** Пользователь переходит на страницу с конкретной задачей, к которой он хочет добавить файл.
2. **Форма добавления файла:** На странице задачи предоставляется форма для загрузки файла, возможно, с полями для указания имени файла или описания.
3. **Выбор файла:** Пользователь выбирает файл на своем устройстве, который он хочет прикрепить к задаче.
4. **Загрузка файла:** После выбора файла пользователь нажимает кнопку "Загрузить" или подтверждает добавление файла. Файл загружается на сервер и связывается с задачей.
5. **Подтверждение добавления:** Пользователю выводится сообщение о успешном добавлении файла к задаче.
6. **Просмотр добавленного файла:** Пользователь может просмотреть добавленный файл на странице задачи, возможно, с предоставленной ссылкой для его скачивания.

## Тема 2 – Интернет-магазин

### Обязательные требования:

1. Разработать базу данных с использованием PostgreSQL, включающую связи:
  - 1-1 (например, пользователь - профиль пользователя)
  - 1-m (например, пользователь - заказы)
  - m-m (например, продукты - категории)
2. Реализовать функционал для:
  - Добавления, редактирования товаров в ассортимент
  - Добавления товаров в корзину
  - Просмотра товаров в корзине
  - Редактирования товаров в корзине
  - Удаления товаров из корзины
  - Просмотра списка всех товаров
  - Переход к оплате (без самой оплаты)
3. Реализовать систему регистрации и аутентификации пользователей (логин).

### Необязательные требования:

1. Реализовать систему прав доступа к различным страницам (например, администратор, пользователь).
2. Реализовать возможность аутентификации через Google.
3. Добавить стилизацию с использованием CSS (Bootstrap) и добавить интерактивности с помощью JavaScript.
4. Добавить функционал отправки уведомлений (например, по электронной почте) о заказах или статусе доставки.

### Регистрация:

1. **Страница регистрации:** Пользователь переходит на страницу регистрации, где ему предоставляется форма для ввода информации.
2. **Заполнение формы:** Пользователь заполняет форму, вводя свои данные, такие как имя пользователя, электронная почта и пароль. Дополнительные поля могут включать имя, фамилию и другие необходимые данные.
3. **Проверка данных:** После заполнения формы приложение проверяет введенные данные на корректность и уникальность (например, проверка на уникальность имени пользователя и электронной почты).
4. **Создание учетной записи:** Если данные прошли проверку успешно, приложение создает новую учетную запись пользователя в базе данных.
5. **Успешная регистрация:** Пользователю выводится сообщение об успешной регистрации и, возможно, предлагается автоматический вход в систему.

### Авторизация:

1. **Страница входа:** Пользователь переходит на страницу входа (логина), где ему предоставляется форма для ввода учетных данных.
2. **Ввод данных:** Пользователь вводит свои учетные данные, такие как имя пользователя (или электронная почта) и пароль.
3. **Проверка учетных данных:** Приложение проверяет введенные учетные данные на соответствие данным, хранящимся в базе данных.
4. **Успешная авторизация:** Если введенные данные верны и учетная запись активна, пользователь успешно аутентифицируется и ему предоставляется доступ к защищенным страницам приложения.

### **Просмотр списка товаров:**

#### **1. На главной странице:**

- Пользователь открывает главную страницу интернет-магазина.
- На этой странице отображается список товаров, доступных для покупки.
- Каждый товар представлен кратким описанием, изображением, ценой и возможной скидкой.

#### **2. Навигация по категориям:**

- Пользователь может использовать меню навигации или фильтры по категориям, чтобы увидеть товары определенной категории.
- Категории могут быть расположены на боковой панели или в верхнем меню.

#### **3. Фильтрация и сортировка:**

- Пользователь может использовать различные фильтры (например, цена, бренд, размер) для настройки списка товаров.
- Также предоставлена возможность сортировки товаров по различным критериям, таким как цена или рейтинг.

### **Добавление товаров в корзину:**

#### **1. Нажатие на кнопку "Добавить в корзину":**

- Пользователь выбирает интересующий его товар и нажимает кнопку "Добавить в корзину", которая находится рядом с ним.

#### **2. Обновление информации о корзине:**

- Система добавляет выбранный товар в корзину пользователя.
- Обновляется информация о корзине, отображая общее количество товаров и общую сумму заказа.

### **Просмотр товаров в корзине:**

#### **1. Переход к корзине:**

- Пользователь нажимает на кнопку "Корзина" или на иконку корзины, чтобы перейти к просмотру содержимого корзины.

#### **2. Отображение товаров:**

- На странице корзины пользователь видит список добавленных товаров с их описанием, ценой, количеством и общей стоимостью.
- Также отображается общая сумма заказа и кнопки для изменения количества товаров или удаления товаров из корзины.

### **Удаление товара из корзины:**

#### **1. Изменение количества товара:**

- Пользователь может уменьшить количество товара или удалить его полностью из корзины, нажав на соответствующие кнопки "Удалить" или "Уменьшить количество".

#### **2. Обновление информации о корзине:**

- После удаления или изменения количества товара система обновляет информацию о корзине, показывая новое общее количество товаров и общую сумму заказа.

### **Переход к оплате:**

#### **1. Переход к странице оформления заказа:**

- После выбора всех необходимых товаров пользователь нажимает кнопку "Оформить заказ" для перехода к странице оформления заказа и выбора способа оплаты.

## **2. Заполнение данных:**

- Пользователь заполняет форму с данными для доставки (например, адрес доставки, контактная информация).
- Также выбирает удобный способ оплаты (например, онлайн-платеж, наличными при получении).

## **3. Подтверждение заказа:**

- После заполнения всех данных пользователь подтверждает заказ, нажимая на кнопку "Оплатить" или аналогичную.
- Система обрабатывает заказ, отправляет подтверждение на электронную почту пользователя и перенаправляет его на страницу с подтверждением заказа.

## **Добавление новых товаров на сайт:**

### **1. Авторизация:**

- Администратор авторизуется на сайте, вводя свои учетные данные (логин и пароль).

### **2. Переход к странице администрирования:**

- После успешной авторизации администратор переходит на страницу администрирования, где есть возможность управлять товарами.

### **3. Добавление нового товара:**

- Администратор выбирает опцию "Добавить товар" и заполняет форму с информацией о новом товаре (название, описание, цена, изображение и другие детали).
- После заполнения формы товар сохраняется в базе данных и становится доступным для просмотра и покупки на сайте.

## Тема 3 – Личный блог

### Обязательные требования:

1. Разработать базу данных с использованием PostgreSQL, включающую связи:
  - 1-1 (например, пользователь - профиль пользователя)
  - 1-m (например, пользователь - записи блога)
  - m-m (например, записи блога - теги)
2. Реализовать функционал для:
  - Создания, редактирования, удаления и чтения одной записи (например, статьи или записи блога).
  - Чтения всех записей (например, списка всех статей или записей блога).
3. Реализовать возможность загрузки файлов, например, изображений для статей блога.
4. Реализовать систему регистрации и аутентификации пользователей (логин).
5. Реализовать функционал для смены пароля пользователем.
6. Реализовать функционал для удаления пользователя.

### Необязательные требования:

1. Реализовать систему прав доступа к различным страницам (например, администратор, автор, гость).
2. Реализовать возможность аутентификации через Google.
3. Добавить стилизацию с использованием CSS (Bootstrap) и добавить интерактивности с помощью JavaScript.
4. Интегрировать приложение с использованием сторонних API (например, социальные сети для автоматической публикации записей).
5. Добавить функционал отправки уведомлений (например, по электронной почте) о новых комментариях или подписках на блог.

### Регистрация:

1. **Страница регистрации:** Пользователь переходит на страницу регистрации, где ему предоставляется форма для ввода информации.
2. **Заполнение формы:** Пользователь заполняет форму, вводя свои данные, такие как имя пользователя, электронная почта и пароль. Дополнительные поля могут включать имя, фамилию и другие необходимые данные.
3. **Проверка данных:** После заполнения формы приложение проверяет введенные данные на корректность и уникальность (например, проверка на уникальность имени пользователя и электронной почты).
4. **Создание учетной записи:** Если данные прошли проверку успешно, приложение создает новую учетную запись пользователя в базе данных.
5. **Успешная регистрация:** Пользователю выводится сообщение об успешной регистрации и, возможно, предлагается автоматический вход в систему.

### Авторизация:

1. **Страница входа:** Пользователь переходит на страницу входа (логина), где ему предоставляется форма для ввода учетных данных.
2. **Ввод данных:** Пользователь вводит свои учетные данные, такие как имя пользователя (или электронная почта) и пароль.
3. **Проверка учетных данных:** Приложение проверяет введенные учетные данные на соответствие данным, хранящимся в базе данных.

4. **Успешная авторизация:** Если введенные данные верны и учетная запись активна, пользователь успешно аутентифицируется и ему предоставляется доступ к защищенным страницам приложения.

#### **Просмотр записей в блогах:**

1. **На главной странице:**

- Пользователь заходит на главную страницу блога.
- На странице отображается список последних опубликованных постов от разных авторов.

2. **Выбор записи:**

- Пользователь может просматривать посты различных авторов и выбирать интересующие его записи для прочтения.
- Для этого пользователь нажимает на заголовок или изображение поста, чтобы открыть его полное содержание.

3. **Отображение записи:**

- После выбора поста открывается страница с его полным содержанием, включая заголовок, текст, изображения и другие мультимедийные элементы.

#### **Создание нового поста в блоге:**

1. **Авторизация:**

- Пользователь авторизуется на сайте, используя свои учетные данные.

2. **Переход к созданию поста:**

- После успешной авторизации пользователь переходит на страницу создания нового поста.

3. **Заполнение данных:**

- Пользователь заполняет форму создания поста, включая заголовок, текст, изображения и другие мультимедийные элементы.
- Также пользователь может выбрать категорию поста и добавить теги для его классификации.

4. **Публикация поста:**

- После заполнения данных пользователь нажимает кнопку "Опубликовать", чтобы разместить свой пост на блоге.

#### **Редактирование и публикация своих постов:**

1. **На странице своих постов:**

- Пользователь переходит на страницу со списком своих опубликованных постов.

2. **Выбор поста для редактирования:**

- Пользователь выбирает пост, который хочет отредактировать, и нажимает на кнопку "Редактировать" рядом с ним.

3. **Редактирование данных:**

- После выбора поста пользователь попадает на страницу редактирования, где может изменить заголовок, текст, изображения и другие данные поста.

4. **Сохранение изменений:**

- После внесения изменений пользователь нажимает кнопку "Сохранить", чтобы сохранить обновленные данные поста.

5. **Публикация изменений:**

- После сохранения изменений пост снова публикуется и становится доступным для просмотра другими пользователями.

#### **Удаление своих постов в блоге:**

1. **На странице со списком своих постов:**



- Пользователь переходит на страницу, где отображаются все его опубликованные посты.
- 2. Выбор поста для удаления:**
    - Пользователь находит пост, который хочет удалить, и выбирает его, например, щелкая по заголовку или изображению.
  - 3. Открытие опций редактирования:**
    - После выбора поста открываются опции редактирования или дополнительные команды, включая "Удалить" или "Редактировать".
  - 4. Нажатие на кнопку "Удалить":**
    - Пользователь нажимает на кнопку "Удалить" или аналогичную, которая активирует процесс удаления поста.
  - 5. Подтверждение удаления:**
    - После нажатия на кнопку "Удалить" появляется модальное окно или подтверждающее сообщение, запрашивающее у пользователя подтверждение удаления поста.
  - 6. Подтверждение действия:**
    - Пользователь подтверждает удаление, например, нажимая на кнопку "Подтвердить" в модальном окне.
  - 7. Удаление поста:**
    - После подтверждения действия пост удаляется из базы данных и больше не отображается на сайте.
  - 8. Обновление страницы:**
    - После удаления поста страница автоматически обновляется или пользователь перенаправляется на страницу со списком оставшихся постов.

## Тема 4 – Агрегатор новостей с помощью RSS лент

### Обязательные требования:

1. Разработать базу данных с использованием PostgreSQL, включающую связи:
  - 1-1 (например, пользователь - профиль пользователя)
  - 1-m (например, ресурс - записи новостей)
  - m-m (например, записи новостей - теги)
2. Реализовать функционал для:
  - Создания, редактирования, удаления и чтения одной записи (например, новости).
  - Чтения всех записей (например, списка всех новостей).
3. Реализовать возможность загрузки файлов, например, изображений для новостей.
4. Реализовать систему регистрации и аутентификации пользователей (логин).
5. Реализовать функционал для смены пароля пользователем.
6. Реализовать функционал для удаления пользователя.

### Необязательные требования:

1. Реализовать возможность аутентификации через Google.
2. Добавить стилизацию с использованием CSS (Bootstrap) и добавить интерактивности с помощью JavaScript.
3. Добавить функционал отправки уведомлений (например, по электронной почте) о новых записях или подписках на новостной блог.

### Регистрация:

1. **Страница регистрации:** Пользователь переходит на страницу регистрации, где ему предоставляется форма для ввода информации.
2. **Заполнение формы:** Пользователь заполняет форму, вводя свои данные, такие как имя пользователя, электронная почта и пароль. Дополнительные поля могут включать имя, фамилию и другие необходимые данные.
3. **Проверка данных:** После заполнения формы приложение проверяет введенные данные на корректность и уникальность (например, проверка на уникальность имени пользователя и электронной почты).
4. **Создание учетной записи:** Если данные прошли проверку успешно, приложение создает новую учетную запись пользователя в базе данных.
5. **Успешная регистрация:** Пользователю выводится сообщение об успешной регистрации и, возможно, предлагается автоматический вход в систему.

### Авторизация:

1. **Страница входа:** Пользователь переходит на страницу входа (логина), где ему предоставляется форма для ввода учетных данных.
2. **Ввод данных:** Пользователь вводит свои учетные данные, такие как имя пользователя (или электронная почта) и пароль.
3. **Проверка учетных данных:** Приложение проверяет введенные учетные данные на соответствие данным, хранящимся в базе данных.
4. **Успешная авторизация:** Если введенные данные верны и учетная запись активна, пользователь успешно аутентифицируется и ему предоставляется доступ к защищенным страницам приложения.

### Добавление источника новостей в ленту:

- После аутентификации пользователь переходит на страницу добавления источника новостей.

- Пользователь вводит URL RSS ленты или выбирает источник из предложенного списка.
- Система проверяет корректность введенного URL и загружает содержимое RSS ленты.
- Пользователь может выбрать интересующие его категории или теги для данного источника новостей и сохранить его в своей ленте.

#### **Просмотр источников новостей:**

- Пользователь переходит на страницу своей ленты, где отображаются все добавленные им источники новостей.
- Для каждого источника новостей показывается его название, описание и последние загруженные новости.
- Пользователь может просматривать новости из разных источников, переключаясь между ними.

#### **Редактирование источников новостей:**

- Пользователь переходит на страницу управления источниками новостей.
- Для каждого источника новостей показывается список категорий или тегов, которые пользователь может выбирать или изменять.
- Пользователь может добавлять или удалять категории или теги, а также изменять настройки источника новостей.

#### **Отображение списка источников новостей:**

- На странице пользователя отображается список всех источников новостей, которые он добавил в свою ленту.
- Каждый источник новостей представлен в виде карточки или элемента списка с названием и кратким описанием.

#### **Выбор источника новостей:**

- Пользователь выбирает интересующий его источник новостей из списка, нажимая на соответствующий элемент.

#### **Загрузка новостей из выбранного источника:**

- После выбора источника новостей система загружает все доступные новости из этого источника.
- Новости могут загружаться постранично или автоматически подгружаться при прокрутке страницы.

#### **Отображение новостей:**

- Полученные новости отображаются на странице пользователя в виде карточек, где для каждой новости показывается заголовок, краткое описание, изображение (если есть), дата публикации и ссылка на полный текст новости.

#### **Навигация по новостям:**

- Пользователь может просматривать новости из выбранного источника, пролистывая их вверх и вниз с помощью скролла или навигационных элементов.
- Для удобства пользователю предоставляются кнопки навигации, позволяющие переходить к предыдущей или следующей странице новостей.

#### **Чтение полной версии новости:**

- Пользователь может прочитать полную версию новости, нажав на заголовок или ссылку на новость.
- При нажатии система перенаправляет пользователя на сайт источника новости, где он может прочитать полный текст.