Проектное задание для студентов, изучающих язык программирования Python, может быть создано на основе разработки информационной системы для учета и управления задачами. Проект включает в себя базовый и продвинутый уровни требований, а также дополнительные пояснения к функциональным требованиям.

Название проекта: "Система управления задачами"

Цель проекта: Создать информационную систему для учета, управления и отслеживания задач.

**Функциональные требования:**

**Базовый уровень:**

* Авторизация и регистрация: Пользователи могут регистрироваться и авторизовываться в системе.
* Добавление задач: Пользователи могут добавлять новые задачи, включая название, описание и срок выполнения.
* Просмотр задач: Пользователи могут просматривать список всех задач с их деталями.
* Изменение статуса задачи: Пользователи могут отмечать задачи как "выполнено".
* Фильтрация задач: Пользователи могут фильтровать задачи по статусу (выполнено/не выполнено) и сроку выполнения.

**Продвинутый уровень:**

* Приоритет задач: Добавьте возможность задавать приоритеты задач (например, низкий, средний, высокий).
* Уведомления о сроках выполнения: Пользователи получают уведомления по электронной почте о приближающихся сроках выполнения задач.
* Комментарии и обсуждение: Пользователи могут добавлять комментарии к задачам и обсуждать их внутри системы.
* Экспорт и импорт задач: Добавьте возможность экспорта и импорта списка задач для обмена данными.

**Бизнес-требования:**

* Система должна быть доступна через веб-интерфейс.
* Данные о задачах и их статусах должны храниться в базе данных.
* Уведомления о сроках выполнения задач должны быть автоматическими и своевременными.
* Система должна обеспечивать безопасность данных и аутентификацию пользователей.

**Пояснения к функциональным требованиям:**

Для реализации уведомлений о сроках выполнения задач через электронную почту, можно использовать библиотеку Python для отправки электронных писем (например, smtplib) и планирование задач (например, с помощью библиотеки schedule).

Для хранения данных о задачах и пользователях можно использовать реляционную базу данных, такую как SQLite или PostgreSQL, и библиотеку Python для работы с базами данных (например, SQLAlchemy).

Для реализации веб-интерфейса можно использовать фреймворк для веб-разработки на Python, такой как Flask или Django.

Для обеспечения безопасности данных и аутентификации пользователей можно использовать библиотеки Python для работы с аутентификацией и авторизацией, такие как Flask-Login или Django Auth.

Этот проект предоставляет студентам возможность практически применить навыки, изученные в рамках курса по Python, и создать полезную информационную систему для управления задачами.