Ниже представлены 6 равнозначных по сложности вариантов заданий для проектной работы в командах по 4–6 человек. Каждое задание предполагает разработку полноценного программного продукта с технической документацией, интерфейсом, базой данных, соблюдением принципов командной разработки (Git, распределение ролей, презентация проекта и т.д.).

Под вариантами указано само задание общее для всех вариантов и всех команд, за доп вопросами обращаться к преподавателю.

**Вариант 1: Система учета студенческого посещения и успеваемости**

**Краткое описание:**

Создать систему, позволяющую преподавателям отмечать посещения, выставлять оценки, а студентам — просматривать свою статистику.

**Основной функционал:**

* Регистрация преподавателей и студентов.
* Создание и ведение списков групп.
* Отметки посещаемости по дням.
* Ведение журнала оценок.
* Отчетность по каждому студенту.

**Вариант 2: Цифровая библиотека**

**Краткое описание:**

Сервис для хранения и выдачи электронных книг с возможностью бронирования и чтения онлайн.

**Основной функционал:**

* Управление пользователями (читатель, библиотекарь).
* Загрузка и категоризация книг.
* Бронирование книг.
* История действий пользователя.
* Рейтинг и отзывы.

**Вариант 3: Приложение для управления задачами команды (Task Manager)**

**Краткое описание:**

Создание приложения по типу Trello для планирования задач внутри команды.

**Основной функционал:**

* Создание проектов и досок.
* Назначение задач на участников.
* Отслеживание статусов задач (To Do, In Progress, Done).
* Система уведомлений.
* Отчет по выполненным задачам.

**Вариант 4: Система бронирования кабинетов и ресурсов в колледже**

**Краткое описание:**

Платформа, с помощью которой преподаватели и студенты могут бронировать аудитории, технику и пр.

**Основной функционал:**

* Просмотр расписания доступных помещений/ресурсов.
* Создание заявки на бронирование.
* Утверждение администратором.
* Автоматическая проверка на конфликты.
* История бронирований.

**Вариант 5: Информационная система техподдержки**

**Краткое описание:**

Система для подачи заявок в техническую поддержку и их отслеживания.

**Основной функционал:**

* Регистрация пользователей и сотрудников ТП.
* Создание заявок с приоритетами.
* Комментарии и статусы заявок.
* Назначение сотрудников.
* Отчеты по заявкам.

**Вариант 6: Онлайн-дневник питания и калорий**

**Краткое описание:**

Личное приложение для учета потребления калорий и составления рациона.

**Основной функционал:**

* Ведение профиля пользователя (вес, рост, цели).
* База продуктов с калорийностью.
* Составление дневного рациона.
* Расчет потребленных калорий.
* Графики прогресса.

**Что необходимо сделать:**

**1. Анализ и планирование**

* Ознакомьтесь с выбранной задачей (один из предложенных 6 вариантов).
* Составьте **Техническое задание (ТЗ)**:
  + Описание предметной области;
  + Требования к функционалу;
  + Структура пользователей;
  + Ограничения.

**2. Проектирование**

* Спроектируйте архитектуру приложения.
* Создайте **ER-диаграмму** базы данных (если она используется).
* Нарисуйте **диаграммы бизнес-процессов или интерфейсов (UI-макеты)**.(диаграмма вариантов использования/use case)
* Пропишите пользовательские сценарии (use cases).

**3. Разработка ПО**

* Выберите стек технологий (например: Python + Flask, Django, Kivy, JavaScript + React, C# и т.п.).
* Реализуйте:
  + Авторизацию пользователей;
  + Основной функционал, соответствующий варианту задания;
  + Интерфейс (графический или веб).
* Ведение проекта в системе контроля версий (GitHub, GitLab и т.д.).

**4. Тестирование**

* Проведите тестирование:
  + Функциональное (работает ли функционал);
  + UI (удобство использования);
  + Проверка валидации данных.
  + Добавить в код, обязательно, обработчик событий на ошибки (try except)

**5. Документация**

* Итоговый пакет должен включать:
  + Техническое задание (ТЗ);
  + Описание архитектуры;
  + Инструкцию по запуску проекта;
  + Презентацию проекта.(короткая на 6-8 слайдов, всей командой рассказать и представить)

**6. Презентация**

* Подготовьте защиту проекта (10–15 минут):
  + Расскажите о цели проекта;
  + Покажите архитектуру и интерфейс;
  + Продемонстрируйте работу приложения;
  + Расскажите о распределении задач;
  + Ответьте на вопросы преподавателя и одногруппников.

**Советы**

* Работайте как настоящая команда: обсуждайте идеи, устраивайте встречи, фиксируйте прогресс.
* Используйте Notion или Google Docs для управления задачами.
* Пишите понятный и читаемый код, комментируйте важные части.(важно не забывать что код должен быть написан по стандарту, чтобы он был читаем и понятен любому члену команды, не забываем оставлять комментарии и прочее)
* Обязательно проверяйте проект на баги и готовьте демонстрацию заранее.

**ПРЕДУПРЕЖДЕНИЕ: ИСПОЛЬЗОВАНИЕ CHAT GPT СТРОГО ЗАПРЕЩЕНО.**

**ЭТО ЗАДАНИЕ ВЫ ВЫПОЛНЯЕТЕ ДЛЯ СЕБЯ, ЧТОБЫ ПОНЯТЬ, КАК РАБОТАЕТСЯ В КОМАНДЕ, РАЗРАБОТАТЬ ПРОЕКТ, ПОУЧАВСТВОВАТЬ В КОМАНДНОЙ РАЗРАБОТКЕ, НЕМНОГО BRAINSHTORM И КОЛЛЕКТИВНАЯ РАБОТА. ВСЯ ВАША РАБОТА ПОТОМ СКАЖЕТСЯ НА ВАШИХ ЭКЗАМЕНАХ, КВАЛИФИКАЦИОННЫХ ЭКЗАМЕНАХ, А ТАКЖЕ НА ДЕМО. ЭКЗАМЕНЕ!!!!!!!!!!!!**