**Assignment Rules & Regulations:**

- **Plagiarism, cheating, or any form of academic dishonesty is strictly prohibited**. All work submitted must be the student's own, and all external sources, such as code snippets, libraries, or academic papers, **must be properly cited**.
- **Requests for extensions must be requested in advance of the deadline** and should be supported by valid and documented reasons, such as medical or family emergencies. Extensions will be granted at the instructor's discretion.
- If you encounter any issues or require clarifications related to the project, promptly **communicate with the instructor or teaching assistant**. **Do not share your project solutions with classmates before the submission deadline**.

---

This document includes project ideas which you should choose from. Details about phase requirements, date of submission and date of discussion are included as well. It is a team project of **exactly three team members**.

## Deadline for teams formation and idea selection:

- Saturday, the 23rd of November, 2024 at 11:59 pm on LMS.

## Deadline for editing team members:

- Saturday, the 4th of December.

## The Phase Submission Deadline:

- Saturday, the 28th of December, 2024 at 11:59 pm on LMS.

## The Phase Discussion:

- Starting 29th of December, during the practical-exam week.



# Project Ideas:

### 1. Library Management System

**Description:**
Create a system to help librarians manage books and their availability. The system should allow users to add books, check their availability, and keep track of borrowed books.
For example, a user should be able to borrow *"Harry Potter"* and return it later. Overdue books should be flagged for the librarian.

**What to Implement:**

- A way to add books with details like title, author, and publication year.
- Search for books by title or author.
- Borrow books, keep track of who borrowed them, and their due date.
- Return books and mark them as available again.
- Save all data (books, borrowers, and due dates) into a file.

**Example Reference:**
Think of a small school library system that tracks 20-50 books and 10-20 students.

---

### 2. Student Grades Management System

**Description:**
Design a program for teachers to record and calculate grades for students in different courses. For example, the system should allow teachers to enter grades for a student named "Ali" in "Math" and calculate his GPA.

**What to Implement:**

- Add students with details like name and ID.
- Add courses and assign grades to students.
- Calculate the GPA for each student.



- Generate a report of all grades for a student in a course.
- Save all data into a file to use again later.

**Example Reference:**
Simulate managing grades for 5-10 students in 3-4 courses, such as Math, English, and Computer Science.

---

### 3. Restaurant Order Management System

**Description:**
Develop a system to manage a restaurant's menu and customer orders. For example, the system should let a customer order "Pizza" and "Soda" and then calculate the total bill.

**What to Implement:**

- Add items to the menu with details like name and price.
- Take customer orders by selecting items from the menu.
- Calculate and display the total bill for an order.
- Track order status (pending, completed).
- Save menu and order data for future use.

**Example Reference:**
Imagine a small restaurant with 10 menu items like Pizza, Burger, Soda, and Coffee.

---

### 4. Event Management System

**Description:**
Create a system to help organizers manage events and attendee registrations. For example, it should allow organizers to add events like a "Coding Workshop" and register attendees for it.



**What to Implement:**

- Add events with details like name, date, and location.
- Register attendees for specific events.
- Display a list of attendees for each event.
- Generate a summary of all events and attendee counts.
- Save event and attendee data to a file.

**Example Reference:**
Manage 3-5 events, such as "Math Club Meetup" or "Science Fair," with 10-15 attendees.

---

**5. Inventory Management System**

**Description:**
Build a system to manage the inventory of a small shop. For example, the system should help track products like "Apples" or "Milk," their quantities, and notify when stock is low.

**What to Implement:**

- Add items to the inventory with details like name, price, and quantity.
- Search for items by name.
- Reduce stock when items are sold.
- Notify when stock is below a certain threshold (e.g., less than 5).
- Save inventory data to a file.

**Example Reference:**
Track 10-15 items like fruits, snacks, and drinks with quantities between 5-20.

---



### 6. Personal Budget Tracker

**Description:**
Develop a program to help users track their income and expenses. For example, users should be able to add "Salary" as income and "Groceries" as an expense, then see how much money they have left.

**What to Implement:**

- Add income and expenses with a description, amount, and date.
- Categorize expenses (e.g., Food, Transport, Entertainment).
- Display total income, total expenses, and remaining balance.
- Generate a monthly summary of spending.
- Save all financial data to a file.

**Example Reference:**
Simulate tracking income like a monthly salary and expenses such as groceries, rent, and movie tickets.

---

### 7. Clinic Appointment Management System

**Description:**
Create a system to help clinic staff manage patient appointments. For example, the system should allow scheduling an appointment for "Dr. Smith" at 10:00 AM for "John Doe."

**What to Implement:**

- Add patients with details like name and contact information.
- Schedule appointments for a specific date and time.
- Update or cancel appointments.
- Display a list of daily appointments.
- Save appointment data to a file.



**Example Reference:**
Manage appointments for a single doctor with 5-10 patients per day.

---

### 8. Hotel Room Booking System

**Description:**
Design a system to manage hotel room bookings. For example, it should allow booking "Room 101" for 2 nights and calculate the cost.

**What to Implement:**

- Add rooms with details like type, price, and availability.
- Book a room and mark it as unavailable.
- Calculate the cost based on the number of nights.
- Display available rooms for a given date.
- Save booking data to a file.

**Example Reference:**
Simulate a small hotel with 5-10 rooms of different types (single, double, suite).

---

### 9. Fitness Center Membership System

**Description:**
Build a system to manage members and their activities at a fitness center. For example, the system should register "Sara" for a "Yoga Class" and track her membership expiration date.

**What to Implement:**

- Add members with details like name and contact information.
- Register members for activities (e.g., Gym, Yoga).
- Track membership expiration dates.
- Generate a report of members and their activities.



- Save membership data to a file.

**Example Reference:**
Manage 10-15 members and 2-3 activities, such as Gym, Swimming, and Yoga.

---

**10. Car Rental Management System**

**Description:**
Create a system to manage car rentals. For example, it should allow booking a "Toyota Corolla" for 3 days and calculate the total rental cost.

**What to Implement:**

- Add cars with details like model, registration number, and price per day.
- Book a car and mark it as unavailable.
- Calculate the cost based on the rental duration.
- Display available cars for a specific date.
- Save booking data to a file.

**Example Reference:**
Simulate a car rental company with 5-10 cars like sedans, SUVs, and vans.

Here are 5 additional project ideas with descriptions tailored for **year 1 Computer Science students**:

---

**11. Pet Adoption System**

**Description:**
Develop a system to manage pets available for adoption at a shelter. For example, the system should let users view pets like "Buddy the Dog" or "Mittens the Cat" and mark them as adopted.

**What to Implement:**



- Add pets with details like name, type (dog, cat, etc.), age, and breed.
- Search for pets by type or name.
- Mark pets as adopted and remove them from the available list.
- Display all available pets and their details.
- Save pet data to a file to keep track of adoptions.

**Example Reference:**
A small shelter managing 5-10 pets with basic details like age, breed, and availability.

---

**12. Online Exam System**

**Description:**
Create a system that allows students to take multiple-choice exams and calculates their scores. For example, students should be able to take a "Python Basics" exam and see their results immediately.

**What to Implement:**

- Add exams with a list of questions and answer choices.
- Allow students to take an exam by answering the questions.
- Automatically calculate and display the score.
- Display a report of all scores for each student.
- Save exam data and student scores to a file.

**Example Reference:**
Include a Python Basics exam with 5-10 questions for testing.

---

**13. E-Commerce Product Catalog System**

**Description:**
Build a system to manage a product catalog for an online store. For example, the system should display products like "Laptop" or "Smartphone" with prices and let users



search for items.

**What to Implement:**

- Add products with details like name, price, and category.
- Search for products by name or category.
- Display all available products in a neat format.
- Mark products as out of stock when sold out.
- Save product data to a file.

**Example Reference:**
Simulate an online store with 10-15 products in categories like Electronics, Clothing, and Home Appliances.

---

**14. Parking Lot Management System**

**Description:**
Design a system to manage cars in a parking lot. For example, the system should allow adding a car with its license plate number and calculate the parking fee based on time spent.

**What to Implement:**

- Add cars entering the parking lot with details like license plate and entry time.
- Remove cars when they leave and calculate the parking fee.
- Display the current number of parked cars and their details.
- Notify if the parking lot is full.
- Save parking data to a file.

**Example Reference:**
Simulate a parking lot with a capacity of 10-15 cars.

---



### 15. Ticket Booking System

**Description:**
Develop a system for booking tickets for events or transportation. For example, it should allow booking a train ticket from "Cairo to Alexandria" and calculate the total cost.

**What to Implement:**

- Add events or routes with details like name, date, and ticket price.
- Book tickets for a specific event or route.
- Track the number of tickets booked and available seats.
- Display booking details and calculate total cost.
- Save ticket and booking data to a file.

**Example Reference:**
Manage ticket bookings for 2-3 events like a concert or train routes with 20-30 seats each.

---

### 16. To-Do List Manager

**Description:**
Create a program to help users manage their daily tasks. For example, users should be able to add tasks like "Complete homework" or "Attend meeting" and mark them as done when finished.

**What to Implement:**

- Add tasks with details like name, priority (e.g., High, Medium, Low), and due date.
- View all tasks and their status (e.g., Pending, Done).
- Mark tasks as complete or delete them when no longer needed.
- Display tasks sorted by priority or due date.
- Save task data to a file for later use.

**Example Reference:**



A user managing 10-15 tasks for a week, with varying priorities and deadlines.

---