**Employee Management System**

**1. Define Requirements:**

* Start by understanding the requirements of the Employee Management System. What features does it need to have? For example, adding new employees, updating employee details, deleting employees, and viewing employee records.
* Let's say our Employee Management System needs to:
  + Allow adding new employees with details such as name, age, and salary.
  + Enable updating employee information.
  + Support deleting employees.
  + Provide functionality to view all employees.

**2. Setup Development Environment:**

* Ensure you have Python installed on your system.
* Choose an Integrated Development Environment (IDE) such as PyCharm.

**3. Plan the Project Structure:**

* Create a project folder named "EmployeeManagementSystem".
* Inside this folder, create Python files for different modules:
  + **main.py**: Entry point of the program.
  + **employee.py**: Module for employee-related functionality.
  + **database.py**: Module for database interactions.

**4. Implement Basic Employee Management Functions:**

* Start by implementing basic functions to perform CRUD operations on employee records.
* In **employee.py**, define functions for CRUD operations:
  + Create a function to add a new employee.
  + Create a function to retrieve employee details.
  + Create a function to update employee information.
  + Create a function to delete an employee.
* Use Python's built-in data structures like lists or dictionaries to store employee records initially.

**5. Organize Code into Modules and Packages:**

* Divide your code into separate modules based on functionality.
  + For example, one module can handle employee management functions, another can handle database interactions, and so on.
  + Import necessary modules in **main.py** and call functions as needed.

**6. Define Employee Class:**

* Define a class named **Employee** to represent an employee.
* Include attributes such as name, age, salary, etc., and methods to manipulate employee data.
* Encapsulate related behaviour within the class.

**7. Connect with RDBMS:**

* Choose an RDBMS (Relational Database Management System) like PostgreSQL.
* Install the necessary database driver/library for Python.
* Establish a connection to the database and create a table to store employee records.
* Modify your functions to interact with the database instead of using in-memory data structures.

**Online Shopping System**

**1. Define Requirements:**

* Define the basic features required for an online shopping system, such as browsing products, adding items to the cart, viewing the cart, and checking out.
* Example: Users should be able to search for products, view product details, add products to the cart, and proceed to checkout.

**2. Setup Development Environment:**

* Install Python and choose an IDE.

**3. Plan the Project Structure:**

* Sketch out the structure of the project, including modules and packages.
* Example:
  + main.py - Main program entry point.
  + products.py - Module for handling product-related functions.
  + cart.py - Module for managing the shopping cart.
  + checkout.py - Module for processing orders.

**4. Implement Basic Shopping Functions:**

* Define functions to handle basic shopping operations.
* Example:
  + display\_products(): Function to display available products.
  + add\_to\_cart(product\_id): Function to add a product to the cart.
  + view\_cart(): Function to view the contents of the cart.
  + checkout(): Function to process the order.

**5. Organize Code into Modules and Packages:**

* Divide the code into separate modules based on functionality.
* Example:
  + products.py can contain functions related to managing products.
  + cart.py can contain functions related to managing the shopping cart.

**6. Define Product Class:**

* Define a class named Product to represent a product.
* Include attributes such as name, price, description, etc.
* Example:

**7. Connect with RDBMS:**

* Choose an RDBMS and set up a database to store product information.
* Establish a connection to the database and create tables.
* Example: Use PSQL and create a table named products with columns for product details.