```
class Employee:
    def __init__(self, emp_id, name, position, salary, email):
        self.emp_id = emp_id
        self.name = name
        self.position = position
        self.salary = salary
        self.email = email
    def update_details(self, name=None, position=None, salary=None,
email=None):
        if name:
            self.name = name
        if position:
            self.position = position
        if salary:
            self.salary = salary
        if email:
            self.email = email
    def to_dict(self):
        return {
            "ID": self.emp_id,
            "Name": self.name,
            "Position": self.position,
            "Salary": self.salary,
            "Email": self.email,
        }
```

```
import csv

class EmployeeManager:
    def __init__(self, filename):
        self.filename = filename
        self.employees = self.load_from_file()

    def load_from_file(self):
        employees = []
        try:
            with open(self.filename, mode='r') as file:
                reader = csv.DictReader(file)
                for row in reader:
                    employees.append(Employee(row["ID"], row["Name"],
row["Position"], row["Salary"], row["Email"]))
        except FileNotFoundError:
            pass  # File doesn't exist yet
        return employees

    def save_to_file(self):
        with open(self.filename, mode='w', newline='') as file:
            fieldnames = ["ID", "Name", "Position", "Salary", "Email"]
            writer = csv.DictWriter(file, fieldnames=fieldnames)
            writer.writeheader()
            for emp in self.employees:
                writer.writerow(emp.to_dict())

    def add_employee(self, emp_id, name, position, salary, email):
        if any(emp.emp_id == emp_id for emp in self.employees):
            return "Employee ID already exists."
```

```
        new_employee = Employee(emp_id, name, position, salary, email)
        self.employees.append(new_employee)
        self.save_to_file()
        return "Employee added successfully."

    def update_employee(self, emp_id, **kwargs):
        for emp in self.employees:
            if emp.emp_id == emp_id:
                emp.update_details(**kwargs)
                self.save_to_file()
                return "Employee updated successfully."
        return "Employee not found."

    def delete_employee(self, emp_id):
        for emp in self.employees:
            if emp.emp_id == emp_id:
                self.employees.remove(emp)
                self.save_to_file()
                return "Employee deleted successfully."
        return "Employee not found."

    def search_employee(self, emp_id):
        for emp in self.employees:
            if emp.emp_id == emp_id:
                return emp.to_dict()
        return "Employee not found."

    def list_all_employees(self):
        return [emp.to_dict() for emp in self.employees]
```

In [ ]:

```
def main():
    manager = EmployeeManager("employees.csv")

    while True:
        print("\nEmployee Data Management System")
        print("1. Add Employee")
        print("2. Update Employee")
        print("3. Delete Employee")
        print("4. Search Employee")
        print("5. List All Employees")
        print("6. Exit")

        choice = input("Enter your choice: ")

        if choice == "1":
            emp_id = input("Enter ID: ")
            name = input("Enter Name: ")
            position = input("Enter Position: ")
            salary = input("Enter Salary: ")
            email = input("Enter Email: ")
            print(manager.add_employee(emp_id, name, position, salary, email

        elif choice == "2":
            emp_id = input("Enter ID of employee to update: ")
            print("Leave fields blank if no change is needed.")
            name = input("Enter new Name: ") or None
            position = input("Enter new Position: ") or None
            salary = input("Enter new Salary: ") or None
            email = input("Enter new Email: ") or None
            print(manager.update_employee(emp_id, name=name, position=positi

        elif choice == "3":
            emp_id = input("Enter ID of employee to delete: ")
            print(manager.delete_employee(emp_id))

        elif choice == "4":
            emp_id = input("Enter ID of employee to search: ")
            result = manager.search_employee(emp_id)
            print(result if isinstance(result, str) else "\n".join(f"{k}: {v

        elif choice == "5":
            employees = manager.list_all_employees()
            if not employees:
                print("No employees found.")
            else:
                for emp in employees:
                    print("\n".join(f"{k}: {v}" for k, v in emp.items()))
                    print("-" * 20)

        elif choice == "6":
            print("Exiting the system. Goodbye!")
            break

        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```

In [ ]:

```
import csv

# Employee Class
class Employee:
    def __init__(self, emp_id, name, position, salary, email):
        self.emp_id = emp_id
        self.name = name
        self.position = position
        self.salary = salary
        self.email = email

    def update_details(self, name=None, position=None, salary=None, email=Nc
        if name:
            self.name = name
        if position:
            self.position = position
        if salary:
            self.salary = salary
        if email:
            self.email = email

    def to_dict(self):
        return {
            "ID": self.emp_id,
            "Name": self.name,
            "Position": self.position,
            "Salary": self.salary,
            "Email": self.email,
        }


# Employee Manager Class
class EmployeeManager:
    def __init__(self, filename):
        self.filename = filename
        self.employees = self.load_from_file()

    def load_from_file(self):
        employees = []
        try:
            with open(self.filename, mode='r') as file:
                reader = csv.DictReader(file)
                for row in reader:
                    employees.append(Employee(row["ID"], row["Name"], row["P
        except FileNotFoundError:
            pass  # File doesn't exist yet
        return employees

    def save_to_file(self):
        with open(self.filename, mode='w', newline='') as file:
            fieldnames = ["ID", "Name", "Position", "Salary", "Email"]
            writer = csv.DictWriter(file, fieldnames=fieldnames)
            writer.writeheader()
            for emp in self.employees:
                writer.writerow(emp.to_dict())

    def add_employee(self, emp_id, name, position, salary, email):
        if any(emp.emp_id == emp_id for emp in self.employees):
            return "Employee ID already exists."
        new_employee = Employee(emp_id, name, position, salary, email)
        self.employees.append(new_employee)
```

```
        self.save_to_file()
        return "Employee added successfully."

    def update_employee(self, emp_id, **kwargs):
        for emp in self.employees:
            if emp.emp_id == emp_id:
                emp.update_details(**kwargs)
                self.save_to_file()
                return "Employee updated successfully."
        return "Employee not found."

    def delete_employee(self, emp_id):
        for emp in self.employees:
            if emp.emp_id == emp_id:
                self.employees.remove(emp)
                self.save_to_file()
                return "Employee deleted successfully."
        return "Employee not found."

    def search_employee(self, emp_id):
        for emp in self.employees:
            if emp.emp_id == emp_id:
                return emp.to_dict()
        return "Employee not found."

    def list_all_employees(self):
        return [emp.to_dict() for emp in self.employees]


# CLI Implementation
def main():
    manager = EmployeeManager("employees.csv")

    while True:
        print("\nEmployee Data Management System")
        print("1. Add Employee")
        print("2. Update Employee")
        print("3. Delete Employee")
        print("4. Search Employee")
        print("5. List All Employees")
        print("6. Exit")

        choice = input("Enter your choice: ")

        if choice == "1":
            emp_id = input("Enter ID: ")
            name = input("Enter Name: ")
            position = input("Enter Position: ")
            salary = input("Enter Salary: ")
            email = input("Enter Email: ")
            print(manager.add_employee(emp_id, name, position, salary, email

        elif choice == "2":
            emp_id = input("Enter ID of employee to update: ")
            print("Leave fields blank if no change is needed.")
            name = input("Enter new Name: ") or None
            position = input("Enter new Position: ") or None
            salary = input("Enter new Salary: ") or None
            email = input("Enter new Email: ") or None
            print(manager.update_employee(emp_id, name=name, position=positi
```

```
        elif choice == "3":
            emp_id = input("Enter ID of employee to delete: ")
            print(manager.delete_employee(emp_id))

        elif choice == "4":
            emp_id = input("Enter ID of employee to search: ")
            result = manager.search_employee(emp_id)
            print(result if isinstance(result, str) else "\n".join(f"{k}: {v

        elif choice == "5":
            employees = manager.list_all_employees()
            if not employees:
                print("No employees found.")
            else:
                for emp in employees:
                    print("\n".join(f"{k}: {v}" for k, v in emp.items()))
                    print("-" * 20)

        elif choice == "6":
            print("Exiting the system. Goodbye!")
            break

        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```