### **1. Description of the Project (5%)**

In this project, we developed a wildfire emergency management system using Python.  
The project’s main objective was to simulate real-world wildfire situations, manage fire alerts, dispatch emergency vehicles, and coordinate evacuation using Queues and Deques.  
It applies Object-Oriented Programming (OOP) concepts like encapsulation, classes, and methods.

### **2. Significance of the Project (5%)**

Wildfires are a serious natural disaster, and fast response saves lives and property.  
This project is useful as it performs key actions like receiving alerts, dispatching resources, and evacuating people which are crucial during wildfires.  
It’s novel because these steps use simple data structures learned in the course (Queue and Deque) and use OOP to organize the code.

### **3. Installation and Usage Instructions (5%)**

* **Installation**:  
  Just copy the Python code and paste it into any ide that supports python (Ex collab)  
  Make sure Python 3 is installed.
* **Usage**:  
  Run the file to simulate receiving fire alerts, dispatching emergency vehicles, and managing evacuation queues.

### **4. Code Structure (5%)**

**Flow:**

FireZone --> FireAlertSystem --> EmergencyResponse --> EvacuationQueue

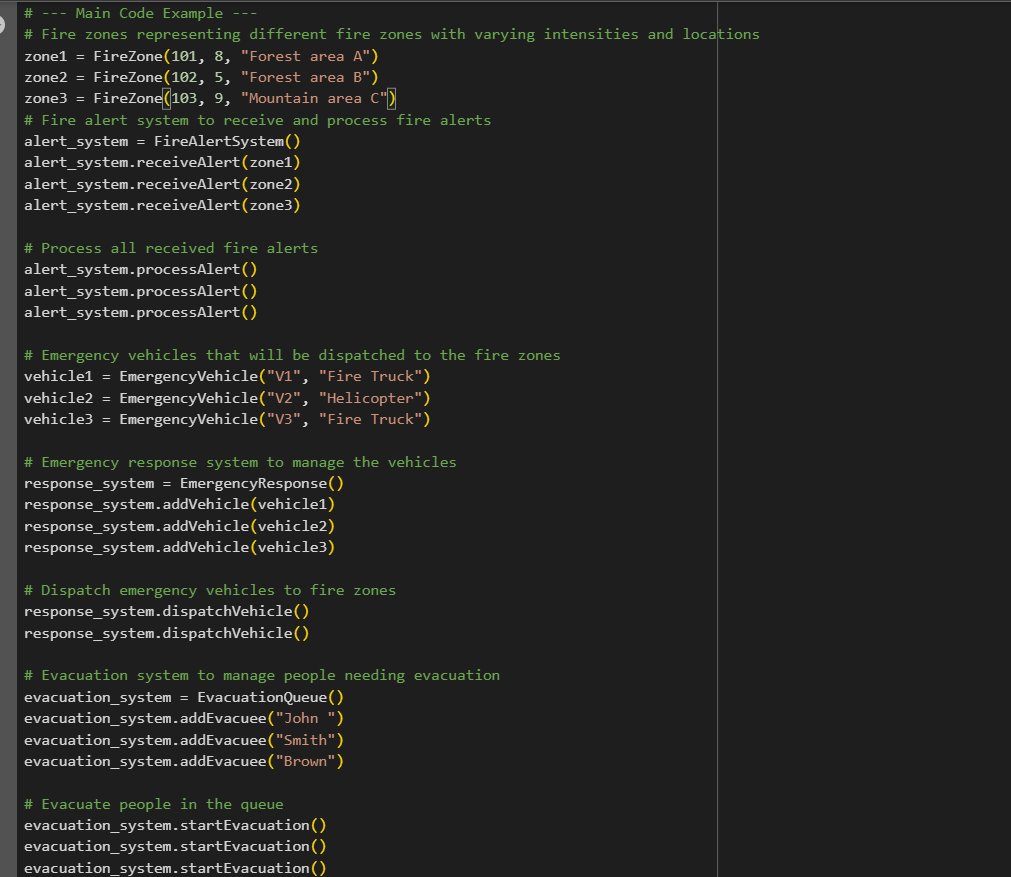
**Main classes:**

* Queue and Deque: Handle the order of alerts, vehicles, evacuees.
* FireZone: Represents each fire alert zone.
* FireAlertSystem: Receives and processes fire alerts.
* EmergencyVehicle and EmergencyResponse: Manage vehicles.
* EvacuationQueue: Manages evacuees.

### **5. Functionalities and Test Results (15%)**

**Functionalities:**

* Add fire alerts to the alert system using Queue.
* Process fire alerts one by one (FIFO order).
* Add emergency vehicles to a Deque.
* Dispatch vehicles (remove from front).
* Add evacuees to evacuation queue.
* Evacuate evacuees in order.



**Test Results:**

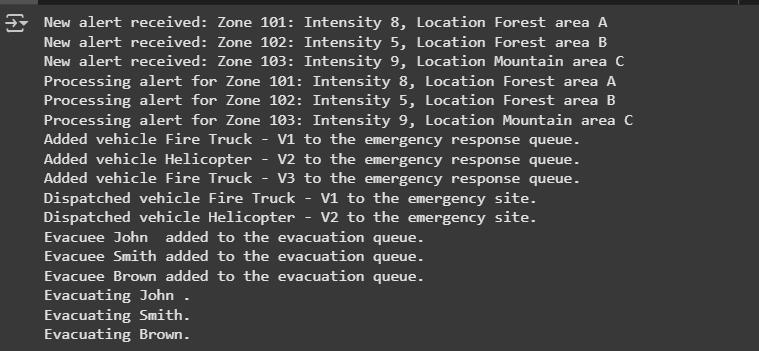
* Fire alerts were processed correctly.
* Emergency vehicles were dispatched properly.
* All evacuees were evacuated without any crash.
* Example print outputs showed successful system operations.

### **6. Showcasing the Achievement of Project Goals (10%)**

The project met its goals by:

* Simulating wildfire alerts and emergency responses.
* Using simple OOP structures like classes and methods.
* Using Queues and Deques to simulate real-world order of operations.

Example output:

****

### **7. Discussion and Conclusions (10%)**

Through this project, we applied core OOP concepts like encapsulation (keeping the queue/deque lists private inside classes).  
I also understood better why using structures like queues is important; they automatically manage correct ordering without manual tracking.  
Limitations: The system is simple and does not handle real-time changes or priorities.  
Future improvements could include adding fire severity-based prioritization or multi-threading for faster simulation.