AshTracker: Wildfire Recovery Assistant - Project Report

1. Description of the Project (5%)

AshTracker is a wildfire recovery assistant that helps users evaluate burned land plots and recommend suitable recovery actions based on terrain, what was burned, and fire severity. It outputs a recovery map and total cost estimate.

Objectives:

- Automate reforestation and rebuilding suggestions.
- Visualize affected areas in a grid.
- Estimate recovery costs based on terrain and damage type.

2. Significance of the Project (5%)

Wildfires leave long-term impacts on the environment and communities. AshTracker helps simulate and plan recovery efforts by applying decision logic using data structures and algorithms. It's novel because it factors in fire severity, terrain compatibility for different trees, and suggests cost-effective options. Unlike basic wildfire simulators, it emphasizes rebuilding strategy and budgeting.

3. Installation and Usage Instructions (5%)

Requirements:

- Python 3.x
- No external libraries needed

To Run:

1. Clone or download the project from GitHub.
2. Open ashtracker.py in any Python IDE.
3. Run the script.
4. Enter number of plots and data when prompted.

4. Code Structure (5%)

Flowchart:

AshTracker: Wildfire Recovery Assistant - Project Report

START -> Get user input -> Suggest recovery -> Display recovery grid -> Print total recovery cost -> END

Main Parts:

- Plot class - stores info about each land plot.
- suggest_recovery() - decision logic.
- get_user_plots() - gets input interactively.
- display_grid() - visual map.
- Summary printing and total cost calculator.

5. Functionalities and Test Results (15%)

Functionality:

- Input Plots: Accepts user data
- Recovery Logic: Suggests action based on terrain + severity
- Grid Display: Shows grid with recovery initials
- Cost Calc: Sums cost for recovery

Test Example:

(0, 0), hill, pine, high -> Suggests 'replant_grass'

2x2 grid -> '[RP][GS]'

Total: $1400

6. Showcasing Achievement of Goals (10%)

- Real data entry simulated through user input
- Outputs recovery strategies for each plot
- Grid helps visualize land recovery distribution
- Total cost gives budget insight
- Shows terrain-specific replanting logic

AshTracker: Wildfire Recovery Assistant - Project Report

7. Discussion and Conclusions (10%)

Issues:

- No real geospatial data
- Text visualization only
- Simplified terrain compatibility

Course Concepts:

- OOP: Plot class
- Functions: Modular recovery logic
- Dictionaries: Terrain rules and costs
- Control structures: if and for

Future Work:

- Add real map visual
- Use CSV/APIs
- Expand tree types

8. Overall Quality of Report and Project (15%)

We developed a complete functional tool applying key programming concepts. The report is structured, includes test results, a flowchart, and explains each part clearly.

9. GitHub (10%)

Ensure the following are uploaded:

- ashtracker.py
- README.md with usage guide
- example_run.txt with sample output

Set the repo to public before submission.