System Requirements Specification Index

For

Hydroponic Farm Monitoring System

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

1 PROJECT ABSTRACT

A small hydroponic farm in the North-East region of India called Karangi Farms needs a system to track plant growth, nutrient levels, and environmental conditions. They require a simplistic interface for logging information for various stages of hydroponic farming. They require separate logs that need to be stored and be accessible in an easy format. Create a python console application that logs information that is needed in simple files utilizing the file handling methodologies commonly used with python.

2 BUSINESS REQUIREMENTS:

Screen Name	Console input screen
Problem Statement	1. Record daily sensor readings in text files 2. Log system activities and alerts 3. Generate reports from historical data 4. Store and retrieve nutrient mixing recipes

3 CONSTRAINTS

3.1 FILE REQUIREMENTS

1. Directory Structure:
2. `sensor_readings.txt`: Daily sensor data
3. `system_log.txt`: Operation logs (append-only)
4. `nutrient_levels.csv`: Nutrient measurements
5. `recipes.txt`: Nutrient mixing recipes

3.2 FILE MODE REQUIREMENTS

1. Read ('r'): For generating reports
2. Write ('w'): For creating new data files
3. Append ('a'): For adding to logs without overwriting
4. Read/Write ('r+'): For updating recipes

4. TEMPLATE CODE STRUCTURE:

1. Basic Functions:

- `read_sensor_data(file_path)` - reads sensor history ('r' mode)
- `save_daily_readings(file_path, data)` - records new readings ('w' mode)
- `log_system_event(file_path, message)` - logs events ('a' mode)

2. Advanced Functions:

- `update_recipe(file_path, recipe_name, new_instructions)` - updates recipes ('r+' mode)
- `backup_data_files(source_dir, backup_dir)` - creates data backups

3. Utility Functions:

- `generate_weekly_report(data_file_path, output_file_path)` - creates reports
- `search_logs(log_file_path, search_term)` - searches logs for specific events

4. Main Program Function:

- `main()` - demonstrates all functions and produces formatted output.

5. EXECUTION STEPS TO FOLLOW:

1. Implement each function using the appropriate file mode
2. Create realistic sample data for testing
3. Demonstrate error handling for common file issues
4. Develop a simple menu-driven interface