🧠 Feature: Intent-Aware Automated Backup Buddy

🎯 Core Idea

Instead of:

❌ Backup every Sunday at 8 PM
❌ Backup everything blindly

You do:

✅ Backup when something IMPORTANT happens
✅ Backup when user intent suggests milestone
✅ Backup when importance score spikes

🏗 High-Level Architecture

```
File Change Monitor
        ↓
Importance Engine
        ↓
Intent Detection Engine
        ↓
Backup Trigger Engine
        ↓
Backup Storage Manager
        ↓
Notification System
```

🛠 Complete Tech Stack

Since you're already building previous features:

🔷 Backend

- FastAPI
- Python

🔷 File Monitoring

- watchdog (real-time file system monitoring)

🔷 Database

- SQLite

🔷 Backup Storage

For Hackathon:

- Local backup folder

Advanced Option:

- Cloud (Google Drive / S3 / IBM COS)

🔷 AI Components

- Importance scoring (from your existing engine)
- Simple NLP keyword intent detection

🧠 Step 1: Detect File Changes

Use `watchdog` to monitor file modifications.

```
from watchdog.observers import Observer
from watchdog.events import FileSystemEventHandler

class FileChangeHandler(FileSystemEventHandler):
    def on_modified(self, event):
        print("File modified:", event.src_path)
```

Now you know when:

- File updated
- File created
- File renamed

🧠 Step 2: Detect Importance Spike

You already calculate:

```
importance_score
heat_score
```

Now detect change:

```
if new_importance_score - old_score > threshold:
    trigger_backup = True
```

Example:

User updates:

```
Final_Report.docx
```

Open count increases
Modification time changes
Importance jumps

System detects spike.

🧠 Step 3: Detect Intent (Milestone Detection)

This is where it becomes intelligent.

Instead of only using numbers — detect semantic intent.

🎯 How to Detect Intent?

1️⃣ Filename Keyword Detection

If filename contains:

- final
- submission
- report
- thesis
- v2
- approved
- signed

Then mark as milestone.

```
keywords = ["final", "submission", "approved", "v2", "report"]
```

```
if any(word in filename.lower() for word in keywords):
    milestone = True
```

2️⃣ Semantic Intent Detection (Advanced)

Convert filename + recent edits to embedding.

Compare with “milestone phrases” embedding:

```
["final submission",
 "project completed",
 "placement resume",
 "hackathon final"]
```

If similarity > threshold → milestone detected.

🧠 Step 4: Backup Trigger Engine

Combine signals:

```
if importance_spike OR milestone_detected:
    backup()
```

🧠 Step 5: Perform Backup

Simple approach:

```
import shutil
import datetime

def backup_file(path):
    timestamp = datetime.datetime.now().strftime("%Y%m%d_%H%M%S")
    backup_path = f"backup/{timestamp}_{os.path.basename(path)}"
    shutil.copy2(path, backup_path)
```

Now you have versioned backup.

🧠 Step 6: Smart Backup Versioning

Instead of saving infinite backups:

Keep:

- Only last 5 versions
- Or only milestone versions

```
if version_count > 5:
    delete_oldest_version()
```

📊 Step 7: Dashboard UI

Show:

File	Last Backup	Reason
Final_Report.docx	2 mins ago	Milestone detected
Resume.pdf	1 day ago	Importance spike

Add notification:

“Auto-backup created: Final_Report.docx”

Judges LOVE visible automation.

🔥 Intelligent Enhancements

🧠 1. Backup Risk Score

If file is:

- Frequently edited
- Very important
- Rarely backed up

Then high risk → backup priority.

🧠 2. Project Milestone Detection

If multiple files in same folder modified together:

→ likely project submission.

Trigger folder backup.

🧠 3. Deadline Mode (Demo Trick)

If filename contains:

- submission
- deadline
- placement
- interview

Increase backup sensitivity.

📦 Folder Structure

```
backup-buddy/
│
├── backend/
│   ├── monitor.py
│   ├── importance_engine.py
│   ├── intent_detector.py
│   ├── backup_engine.py
│   ├── database.py
│
├── backups/
│
├── frontend/
│   ├── BackupDashboard.jsx
```

🚀 Complete Execution Plan

Phase 1 – Basic Version

- File monitoring
- Importance spike detection
- Simple auto backup

Phase 2 – Intent Detection

- Filename keyword detection
- Milestone-based triggers

Phase 3 – Intelligent Versioning

- Smart retention policy
- Risk-based backup frequency

🧠 Why This Is Hackathon-Winning

Because it solves real pain:

Students forget to backup before:

- Submission
- Placement
- Hackathon

Your system prevents disaster.

🏆 Combine With Previous Features

Now your system becomes:

```
Smart Search
Memory Engine
Compression Optimizer
Backup Buddy
```

This is a complete:

🧠 AI Personal Memory Operating System

🎯 Killer Presentation Line

Say this:

“Traditional backup systems run on schedule. Our system runs on intent.”

That line hits HARD.

💡 Practical Hackathon Advice

You do NOT need real-time OS integration.

For demo:

- Simulate file edits
- Simulate importance spike
- Show auto-backup happening

Focus on:

Visual proof + logic explanation.

🧠 Final Recommended Stack For YOU

```
FastAPI
SQLite
Watchdog
Shutil
SentenceTransformers (optional)
React + Chart.js
```

Keep it modular.
