**AI Project Submission – Phase 3: Final Report Template**

**1. Project Title**  
  
Intelligent Event Reminder Bot – Rules + Time Triggers

*Must match previous submissions.*

**2. Team Members**  
  
Name: Muhammad Jawad Ali  
Registration No.: F2022266686  
Section: V8  
Email: f2022266686@gmail.com

| **Name** | **Registration No.** | **Section** | **Email** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

**3. Abstract**  
  
This project aimed to create an intelligent reminder bot capable of notifying users about tasks or events based on time and rule-based logic. The system used Python, schedule-based logic, and voice alerts to assist users in organizing their day. The final outcome was a console-based bot that accepts manual reminders and alerts the user via terminal and text-to-speech.

*A short summary of the entire project – its aim, AI method used, and key outcomes.*

**4. Problem Statement Recap**  
  
Users often forget important tasks due to their busy lifestyles. There is a need for an AI-driven assistant to help users stay on track through personalized reminders.

*Summarize the original problem and its relevance or application.*

**5. AI Techniques Used**  
  
- Rule-Based Systems  
- Scheduling Algorithms (via schedule module)  
- Text Extraction using Regular Expressions  
- Text-to-Speech (for output)

*Explain what AI methods you implemented.*  
Examples:

* Rule-Based Systems
* A\* Search Algorithm
* Genetic Algorithm
* Constraint Satisfaction
* Utility Functions, etc.

**6. System Architecture**  
  
Input (User Reminder Text + Time + Day) → Regex + Schedule Module → Reminder Scheduled → Voice + Console Alert  
Modules used: re, schedule, pyttsx3

*Describe system flow using text and optionally a diagram.*

* Inputs → AI Component → Output/Decision
* Include modules or functions used

**7. Implementation Details**  
  
- User provides reminder message, time (HH:MM), and optional day  
- Regex extracts time and day from input  
- Reminder scheduled using the schedule module  
- Voice alert generated using pyttsx3  
- Background thread continuously checks for reminders

*What was implemented in code?*

* Logic rules
* Pathfinding steps
* Constraints
* Decision tree branches, etc.

**8. Sample Inputs and Outputs**  
  
Example 1 – Daily Reminder  
📝 Reminder: Drink water  
⏰ Time: 14:00  
📅 Day: [blank]  
✅ Output: Daily reminder set at 14:00  
🔔 Reminder: Drink water  
🗣️ Voice: 'Drink water'  
  
Example 2 – Weekly Reminder  
📝 Reminder: Class  
⏰ Time: 08:00  
📅 Day: Monday  
✅ Output: Reminder set for every Monday at 08:00  
🔔 Reminder: Class  
🗣️ Voice: 'Class'

*Provide screenshots, tables, or output samples from your final system.*

**9. Evaluation**  
  
- Accuracy: 100% for correctly formatted inputs  
- Performance: Lightweight and responsive  
- Rule Coverage: Handles both daily and weekly reminders  
- Validated with manual test cases

*How well does your system perform?*

* Accuracy, Efficiency, Correctness, Rule Coverage
* Manual or test case validation

**10. Challenges Faced and Resolved**  
  
- GUI (tkinter) not supported in all environments → Shifted to console-based interface  
- Voice library compatibility managed using pyttsx3  
- NLP feature (spaCy) was simplified to regex for compatibility

*List final development or logic challenges and how they were resolved.*

**11. Learning Outcomes**  
  
- Learned scheduling logic with Python  
- Hands-on with text-to-speech module  
- Practical understanding of rule-based systems

* What did you learn from this project?
* Any concepts clarified during implementation?

**12. Future Improvements (Optional)**  
  
- Add GUI using tkinter or PyQt  
- Integrate NLP using spaCy or transformers  
- Save/load reminders from local storage or cloud

* How would you extend or improve the project?

**13. Final Timeline Review**  
  
Week Task Status  
3–4 Design + initial implementation ✅ Done  
5–6 Logic core + testing ✅ Done  
7–8 Interface + polishing ✅ Done  
9 Final review & report ✅ Done

| **Week** | **Task** | **Status** |
| --- | --- | --- |
| 3–4 | Design + initial implementation | ✅ Done |
| 5–6 | Logic core + testing | ✅ Done |
| 7–8 | Interface + polishing | ✅ Done / ❌ Not Done |
| 9 | Final review & report | ✅ Done |

**Submission Guidelines:**

* Submit as a typed and printed document.
* Attach working code and demo video if required.
* Ensure alignment with previously approved objectives.