Engagement Insight Engine

2. Objective

Build an AI-based microservice that:

- Analyzes user profile strength, activity, and peer comparison
- Suggests **profile improvement nudges** (glow-up advisor)
- Detects **event FOMO signals** based on peer/buddy actions (event radar)
- Is configurable, deterministic, offline, and served via **FastAPI**

3. Functional Requirements

A. Input

- **Endpoint**: `POST /analyze-engagement`
- **Payload**:

```
json

{
  "user_id": "stu_7023",
  "profile": {
    "resume_uploaded": false,
    "goal_tags": ["GRE", "data science"],
    "karma": 190,
    "projects_added": 0,
    "quiz_history": ["aptitude", "python"],
    "clubs_joined": [],
    "buddy_count": 3
  },
  "activity": {
    "login_streak": 2,
    "posts_created": 1,
    "buddies_interacted": 0,
    "last_event_attended": "2024-06-30"
  },
  "peer_snapshot": {
    "batch_avg_projects": 2,
    "batch_resume_uploaded_pct": 84,
    "batch_event_attendance": {
      "startup-meetup": 5,
      "coding-contest": 9
    },
    "buddies_attending_events": ["coding-contest"]
  }
}
```

B. Output

json

```
{
  "user_id": "stu_7023",
  "nudges": [
    {
      "type": "profile",
      "title": "84% of your peers have uploaded resumes. You haven’t yet!",
      "action": "Upload resume now",
      "priority": "high"
    },
    {
      "type": "event",
      "title": "3 of your buddies are joining 'Coding Contest'",
      "action": "Join the event",
      "priority": "medium"
    }
  ],
  "status": "generated"
}
```

4. AI Model Requirements

A. Model Type

Use lightweight rule-based ML hybrid:

- Rule layer: Compare user stats to peer group
- AI layer:
    - Classifier to predict **nudging probability**
    - `RandomForestClassifier` or `LogisticRegression` on simulated behavior data

B. Training Dataset (Simulated)

Create 500+ rows of:

```
json

{
  "features": {
    "resume_uploaded": false,
    "karma": 150,
    "batch_resume_uploaded_pct": 90,
    "event_fomo_score": 3
  },
  "label": {
    "should_nudge_resume": 1,
    "should_nudge_event": 1
  }
}
```

5. Nudge Generation Logic

Profile Nudges:

Condition	Nudge Example
Resume missing + 80% peers have uploaded	Suggest resume upload
Projects = 0 + batch avg > 2	Suggest adding projects
Not taken quiz for 7+ days	Push “Take 2-question quiz today”

Event Nudges:

Condition	Nudge Example
3+ buddies joining an event	"Your friends are joining..."
10+ same batch users joined	"Your peers are in!"
User inactive for 5+ days	Trigger comeback nudge via event invite

6. Config Requirements

`config.json`:

json

```json
{
  "profile_rules": {
    "resume_threshold": 0.7,
    "projects_avg_threshold": 2,
    "quiz_idle_days": 7
  },
  "event_rules": {
    "buddy_attendance_trigger": 2,
    "batch_attendance_trigger": 10
  },
  "priority_labels": {
    "resume": "high",
    "project": "medium",
    "quiz": "low",
    "event_fomo": "medium"
  }
}
```

7. API Endpoints

Method	Endpoint	Description
POST	`/analyze-engagement`	Returns nudges for profile & events
GET	`/health`	`{ "status": "ok" }`

GET	`/version`	`{ "version": "1.0.0" }`

8. FastAPI Requirement

- Use **FastAPI**
- I/O defined with **Pydantic** models
- Serve with **Uvicorn on port 8000**
- Expose docs via `/docs`

9. Testing & Validation

Scenario	Expected Outcome
Resume not uploaded, high batch resume %	Resume nudge shown
Buddies joining event	Event nudge shown
User up-to-date profile + no FOMO	No nudges returned

Minimum 10 profiles × 3 batch peer sets.

10. Deliverables

- Code:
    - `main.py`, `nudge_engine.py`, `config.json`, `model.pkl`
- Simulated `peer_snapshot.json`
- Sample input/output logs
- `README.md` with API guide
- Minimum 5 test scenarios

11. Timeline (6 Weeks)

Week	Tasks
Week 1	Simulate profiles + peer data
Week 2	Train nudging model
Week 3	Rule + AI blend scoring logic
Week 4	FastAPI interface
Week 5	Generate output messages + test cases
Week 6	Docker + README polish

12. Constraints

- Fully **offline**
- No cloud dependencies
- Max 3 nudges per day
- Deterministic rules must override ML if conflict

13. Deployment Expectations

```
docker build -t engagement-insight-engine .
docker run -p 8000:8000 engagement-insight-engine
```