INTELLIGENT TIMETABLE – 24 HOUR HACKATHON ROADMAP

Stack: Python, Flask, SQLite, SQLAlchemy, OR-Tools (CP-SAT)

0. PREP (30–45 min)

- Create project folder and virtualenv
- Install packages
- Create project structure

1. BASIC FLASK APP + AUTH

- Setup Flask app
- Create User model
- Implement login/logout
- Create basic templates
- Dashboard route

2. DATA MODELS

- Room, Batch, Subject, Faculty, Timeslot
- Timetable and TimetableEntry models
- Run db.create_all()

3. SEED DATA

- Admin user
- Rooms, batches, subjects, faculties
- FacultySubject mapping
- Timeslot generation

4. OR-TOOLS SCHEDULER

- Build ClassRequirement list
- Create CP-SAT variables
- Add constraints:
- One class per slot
- No faculty clashes
- No room clashes
- No batch clashes
- Capacity constraints
- Solve and store results in DB

5. GENERATE + VIEW ROUTES

- /generate form
- Call generate_timetable()
- /timetable/ viewer with grid layout

6. CLASH DETECTION & METRICS

- Check duplicated batch/faculty/room in same slot
- Highlight conflicts
- Metrics: room utilization, faculty load

7. MANUAL RESCHEDULING (OPTIONAL)

- Suggest alternate slots
- Update timetable entries

8. UI POLISH + DEMO PREP

- Cleanup UI
- Prepare demo dataset
- Demo script

9. PRIORITY ORDER

Must-have:

- Login
- Seeded data
- Timetable generation
- Timetable view

Optional:

- Clashes, metrics
- Rescheduling