Major work splits

1. Frontend – task level – exhaustive
2. Backend (database building and querying) – task level – exhaustive
3. Composing all independent builds to single super web – task level – Suicidal
4. Rag agent and analytics UI and team regulation – task level – medium (push the team to its boundaries)
5. Clerical work and testing – task level –   exhaustive (single person writes entire paper with citations) (AND ppt)

Frontend Flow

1. Login page
   a. Admin / Faculty Login
   b. Forgot password
2. Faculty page
   a. Dashboard
      i. Courses handled
      ii. Timetable (simple display)
      iii. Academic calendar (simple display)
      iv. Performance tab (select course)
      v. Notification space (Announcement)
      vi. Faculty Profile
   b. Course Handled
      i. Student List
      ii. Assignment Deadlines
      iii. Resources
      iv. Grade Viewing System
      v. Forum (Optional) (rag will reply)
   c. Performance Tab (All visual analytics):
      i. Attendance
      ii. Average grade (rag querying and display)
      iii. Assignment wise Performance analysis (rag querying)
   d. Faculty Profile
      i. Basic things.

Backend Flow

1. Login
   a. Password
   b. Email
   c. role
2. Faculty Details
   a. Basic Info
3. Course List
   a. Course name
   b. Class
   c. Course Resources (As drive link)
   d. Course id
   e. Faculty name
4. class list (many tables):
   a. Class name
   b. Class id
5. Student database
   a. Student name
   b. Roll no
   c. Class id
6. Attendance table
   a. Attendance_id
   b. Student_id
   c. Course_id
   d. Date
   e. Status - (present,absent,late)
7. Assignment table
   a. Assignment id
   b. Course id
   c. Title
   d. Deadline

e. Submission link
8. Assignment submission table
   a. Submission _id
   b. Assignment_id
   c. Student_id
   d. Submission_date
   e. File_link
9. Grades table
   a. Student_id
   b. Course_id
   c. Assignment_id
   d. Grade
10. Course deadline table (optional)
    a. Date
    b. Deadline name
    c. Course id

Connection Flow:

API endpoints must be cross verified and connected

RAG Agent:

1. Use ollama nothing much. (No need gemini as of now. While proposing we can say administrations wish on llm)
2. If resources are given in the same directory in pdf format, we can easily implement it.
3. Pass sql access to LLM.
4. Retrieve from LLM and post it in UI.
5. Force and ensure the team does the work.
6. Complete the task and join in backend development.

Clerical Works AND Testing:

1. 4 Documentations left.
   a. Sprint Reports
2. Conf. Paper (full writing).
3. Upcoming presentations making.
4. Final presentation PPT.
5. Case Study PPT.

6. Conference finding.
7. Testing and test report.

Common Works:

1. Check and read the paper
2. Give inputs to the paper
3. Supervise Clerical Works