Task Description!

- **Designing the Schema** [Maryam & Maria; both met to discuss]**:**
  - Plan entities like Tutors, Students, Availability, and Bookings.
  - Define relationships between these entities (e.g., many-to-many, one-to-many).
  - Incorporate fields for:
    - Tutor and student details.
    - Time slots.
    - Booking details and statuses.
    - Support for time zones.
  - ✔ Complete by Monday
- **Listing the APIs** [Maryam & Maria; split between both of us; 19 total]**:**
  - Identify all necessary RESTful APIs for the scheduling system.
  - Include key features such as:
    - Tutor availability management.
    - Student booking functionality.
    - Conflict management.
    - Notifications for bookings and availability updates.
  - Document each API with:
    - Endpoint URL and HTTP method.
    - Purpose.
    - Request parameters/body.
    - Response format.
  - ✔ Complete by Tuesday
- **Building the APIs** [Maryam & Maria; 3 API's each]**:**
  - Implement 6 APIs based on priority:
    - Adding or updating tutor availability.
    - Booking a time slot.
  - Ensure the APIs follow best practices for RESTful design:
    - Clear resource naming.
    - Use proper HTTP methods (GET, POST, PUT, DELETE).

        - Standardize response formats (e.g., JSON).
    - ✔ Complete by Tuesday
- **Deployment & Testing** [Maryam & Maria]**:**
    - Create a Flask server.
    - Configure PostgreSQL as the database backend.
    - Write unit tests for the APIs to ensure functionality and robustness.
    - Perform end-to-end testing.
    - Deploy the Flask server locally for demonstration.
    - ✔ Complete by Wednesday
- **Video Presentation & Completion** [Maryam & Maria]**:**
    - Complete the READme file
    - Prepare script
    - Film/Record video
    - ✔ Complete by Wednesday