**1. Use Case Diagram**

* **Actors**: Users (students), Admins
* **Use Cases**:
  + Register/Login
  + Schedule AI Mock Interviews
  + Take Tests
  + View Roadmaps
  + Get Job Recommendations
  + Create/Update Resume
  + Access Resources

**2. Flowchart**

* **Mock Interview Scheduling**:
  1. User logs in
  2. User selects "Schedule Mock Interview"
  3. User selects interview type and preferred date/time
  4. System schedules the interview and sends confirmation
* **Resume Creation**:
  1. User logs in
  2. User selects "Create Resume"
  3. User fills in details (education, experience, skills)
  4. System generates ATS-friendly resume
  5. User downloads or prints resume

**3. Entity-Relationship Diagram (ERD)**

* **Entities**:
  + User (UserID, Name, Email, Password, ProfileDetails)
  + Interview (InterviewID, Type, DateTime, Status, UserID)
  + Test (TestID, Type, Questions, UserID, Score)
  + Roadmap (RoadmapID, Domain, Steps, Resources)
  + Job (JobID, Title, Description, Requirements, Recommendations)
  + Resume (ResumeID, UserID, Content, ATS\_Friendly)
* **Relationships**:
  + User schedules Interview
  + User takes Test
  + User follows Roadmap
  + User receives Job Recommendations
  + User creates Resume

**4. Class Diagram**

* **Classes**:
  + User
    - Attributes: UserID, Name, Email, Password, ProfileDetails
    - Methods: Register(), Login(), UpdateProfile()
  + Interview
    - Attributes: InterviewID, Type, DateTime, Status
    - Methods: Schedule(), Cancel(), UpdateStatus()
  + Test
    - Attributes: TestID, Type, Questions, Score
    - Methods: Start(), Submit(), Evaluate()
  + Roadmap
    - Attributes: RoadmapID, Domain, Steps, Resources
    - Methods: View(), Update(), Follow()
  + Job
    - Attributes: JobID, Title, Description, Requirements
    - Methods: Recommend(), Apply()
  + Resume
    - Attributes: ResumeID, Content, ATS\_Friendly
    - Methods: Create(), Update(), Export()

**5. Sequence Diagram**

* **Mock Interview Scheduling**:
  1. User initiates scheduling request
  2. System displays available slots
  3. User selects slot
  4. System confirms schedule and updates database
  5. System sends confirmation to user

**6. Activity Diagram**

* **Creating a Resume**:
  1. User logs in
  2. User selects "Create Resume"
  3. User enters personal details
  4. User adds education and experience
  5. System formats content into ATS-friendly format
  6. User reviews and finalizes
  7. User downloads the resume

**7. Deployment Diagram**

* **Components**:
  + Web Server (Handles requests and serves content)
  + Application Server (Processes business logic)
  + Database Server (Stores user data, interview schedules, tests, etc.)
  + API Server (Handles job recommendations, external integrations)

**8. State Diagram**

* **User Account State**:
  + States: Registered, Logged In, Active, Inactive, Deleted
  + Transitions: Register → Login → Active → Logout → Inactive/Deleted

**9. Component Diagram**

* **Components**:
  + User Interface (Frontend for user interactions)
  + Mock Interview Service (Schedules and manages interviews)
  + Test Service (Manages test creation, taking, and evaluation)
  + Roadmap Service (Provides learning path and resources)
  + Job Recommendation Service (Suggests jobs based on profile)
  + Resume Service (Generates ATS-friendly resumes)

**10. Architecture Diagram**

* **High-Level View**:
  + User Interface
    - Frontend (React, Angular)
  + Backend
    - API (Node.js, Django)
    - Business Logic (Python, Java)
    - Database (SQL/NoSQL)
  + External Services
    - Job APIs
    - Resume Parsing APIs
  + Infrastructure
    - Cloud Hosting (AWS, Azure)

**11. Data Flow Diagram (DFD)**

* **Data Flow**:
  1. User submits resume details
  2. System processes data and formats resume
  3. System stores resume in the database
  4. User receives/downloads the final resume

**12. Wireframes**

* **Pages**:
  + Home Page
  + Dashboard (Mock Interviews, Tests, Roadmaps)
  + Resume Builder Page
  + Job Recommendations Page
  + Profile Page