Here’s how the tasks can be broken down sprint-wise for each **user story** by team. Each sprint specifies **developer tasks**, **UI design tasks**, and **testing tasks** to attain the user story.

**Team 1: Authentication and Registration Module**

**Sprint 1: Registration/Login UI**

1. **Developer Tasks:**
   * Create registration and login components with Angular forms.
   * Implement form controls for email, password, and role selection.
2. **UI Design Tasks:**
   * Design responsive layouts for registration and login pages.
   * Use placeholders like "Enter email" and "Enter password."
   * Add role dropdown (Admin, Instructor, Candidate).
3. **Tester Tasks:**
   * Verify UI responsiveness on multiple devices.
   * Check form validations (e.g., required fields, email format).

**Sprint 2: Credentials Verification**

1. **Developer Tasks:**
   * Implement backend API for verifying credentials against the database.
   * Connect login page with the backend for verification.
2. **UI Design Tasks:**
   * Display error messages for invalid credentials.
3. **Tester Tasks:**
   * Test login with valid and invalid credentials.
   * Verify API calls and responses.

**Sprint 3: Role-Based Redirection and Logout**

1. **Developer Tasks:**
   * Implement role-based navigation for Admin, Instructor, and Candidate dashboards.
   * Add secure logout functionality.
2. **UI Design Tasks:**
   * Create dashboards for all roles (mock data is sufficient at this stage).
3. **Tester Tasks:**
   * Test redirection logic based on roles.
   * Verify session termination upon logout.

**Sprint 4: Password Reset and Error Handling**

1. **Developer Tasks:**
   * Implement password recovery and reset APIs.
   * Add error-handling logic for unauthorized access.
2. **UI Design Tasks:**
   * Design password reset and error pages.
3. **Tester Tasks:**
   * Test reset and recovery workflows.
   * Verify error messages for incorrect actions.

**Team 2: Course Module**

**Sprint 1: Create Courses**

1. **Developer Tasks:**
   * Create course component with forms for name, technology, and status.
   * Implement backend API for course creation.
2. **UI Design Tasks:**
   * Design forms for adding courses.
3. **Tester Tasks:**
   * Verify course creation with valid and invalid inputs.
   * Test API integration for course creation.

**Sprint 2: Update and Delete Courses**

1. **Developer Tasks:**
   * Add update and delete functionality.
   * Implement APIs for updating and deleting courses.
2. **UI Design Tasks:**
   * Add update and delete buttons to the course list.
3. **Tester Tasks:**
   * Test update and delete operations.
   * Ensure changes reflect correctly in the UI.

**Sprint 3: Filters and Course Details**

1. **Developer Tasks:**
   * Implement filters by technology, instructor, and status.
   * Add a detailed view for enrolled courses.
2. **UI Design Tasks:**
   * Create UI for filters and detailed course views.
3. **Tester Tasks:**
   * Verify filter functionalities with multiple datasets.
   * Test detailed views for enrolled courses.

**Sprint 4: Integration**

1. **Developer Tasks:**
   * Integrate the module with authentication and other modules.
   * Ensure all API calls are functional.
2. **UI Design Tasks:**
   * Align UI consistency with other modules.
3. **Tester Tasks:**
   * Perform end-to-end testing for the course module.

**Team 3: Instructor Module**

**Sprint 1: Create Instructors**

1. **Developer Tasks:**
   * Create forms for adding instructors.
   * Implement backend APIs for instructor creation.
2. **UI Design Tasks:**
   * Design a user-friendly form for adding instructors.
3. **Tester Tasks:**
   * Test instructor creation with various inputs.
   * Verify database entries.

**Sprint 2: Update and Delete Instructors**

1. **Developer Tasks:**
   * Add update and delete functionality.
   * Implement APIs for these actions.
2. **UI Design Tasks:**
   * Add UI elements for update and delete.
3. **Tester Tasks:**
   * Test update and delete functionalities.
   * Ensure no stale data remains after deletion.

**Sprint 3: Filters and Views**

1. **Developer Tasks:**
   * Implement filters for technology and rating.
   * Allow instructors to view assigned courses.
2. **UI Design Tasks:**
   * Create filter UI and assigned courses views.
3. **Tester Tasks:**
   * Test filter functionalities and assigned course views.
   * Validate role-based data visibility.

**Sprint 4: Integration**

1. **Developer Tasks:**
   * Integrate instructor module with authentication and course modules.
   * Ensure seamless API communication.
2. **UI Design Tasks:**
   * Standardize UI elements across modules.
3. **Tester Tasks:**
   * Conduct end-to-end testing for instructor-related features.

**Team 4: Candidate Module**

**Sprint 1: Candidate Registration**

1. **Developer Tasks:**
   * Build registration form with validation for email, phone, and password.
   * Implement backend for storing candidate data.
2. **UI Design Tasks:**
   * Design a self-registration page with required validations.
3. **Tester Tasks:**
   * Test registration form for valid and invalid inputs.
   * Verify duplicate prevention mechanisms.

**Sprint 2: Update and Delete Candidate Profiles**

1. **Developer Tasks:**
   * Implement update and delete functionality with backend APIs.
2. **UI Design Tasks:**
   * Add update and delete options on the profile page.
3. **Tester Tasks:**
   * Test profile update and deletion processes.

**Sprint 3: View and Filter Candidates**

1. **Developer Tasks:**
   * Allow viewing all candidates for Admin and Instructor.
   * Implement filters by course and technology.
2. **UI Design Tasks:**
   * Create UI for candidate filters.
3. **Tester Tasks:**
   * Test filter functionalities for accuracy.

**Sprint 4: Integration**

1. **Developer Tasks:**
   * Integrate candidate module with course and instructor modules.
2. **UI Design Tasks:**
   * Ensure UI consistency across modules.
3. **Tester Tasks:**
   * Perform end-to-end testing for candidate features.

**Team 5: Assessment and Grading Module**

**Sprint 1: Schedule Assessments**

1. **Developer Tasks:**
   * Create backend logic for assessment scheduling.
   * Build UI for scheduling assessments.
2. **UI Design Tasks:**
   * Design forms for scheduling assessments.
3. **Tester Tasks:**
   * Verify assessment scheduling process.

**Sprint 2: MCQs and Instructions**

1. **Developer Tasks:**
   * Implement JSON-based data structure for MCQs.
   * Display MCQs with instructions on the UI.
2. **UI Design Tasks:**
   * Design a visually appealing MCQ page.
3. **Tester Tasks:**
   * Test MCQ functionality with various datasets.

**Sprint 3: View Scores**

1. **Developer Tasks:**
   * Implement APIs to calculate and display scores.
2. **UI Design Tasks:**
   * Create a score summary page.
3. **Tester Tasks:**
   * Verify score calculations and display accuracy.

**Sprint 4: Integration**

1. **Developer Tasks:**
   * Integrate the assessment module with course and candidate modules.
2. **UI Design Tasks:**
   * Standardize UI components.
3. **Tester Tasks:**
   * Perform end-to-end testing.

**Team 6: Helpdesk Module**

**Sprint 1: Create Tickets**

1. **Developer Tasks:**
   * Implement ticket creation functionality.
2. **UI Design Tasks:**
   * Design a form for creating tickets.
3. **Tester Tasks:**
   * Test ticket creation with various inputs.

**Sprint 2: Respond to Tickets**

1. **Developer Tasks:**
   * Add response functionality for tickets.
2. **UI Design Tasks:**
   * Display ticket responses in a user-friendly manner.
3. **Tester Tasks:**
   * Test ticket response workflows.

**Sprint 3: View and Filter Tickets**

1. **Developer Tasks:**
   * Implement ticket filtering by status and course.
2. **UI Design Tasks:**
   * Design a dashboard for viewing and filtering tickets.
3. **Tester Tasks:**
   * Verify filtering functionalities.

**Sprint 4: Integration**

1. **Developer Tasks:**
   * Integrate helpdesk with other modules.
2. **UI Design Tasks:**
   * Align UI with other modules.
3. **Tester Tasks:**
   * Conduct end-to-end testing.