

Assignment of responsibilities for each team-member of the team:

```
/project-root
  /backend
    /servlets
      UserServicelet.java      // Handle user-related API requests (HTTP Requests → Servi
      TeamServlet.java         // Handle team-related API requests
      ProcessServlet.java      // Handle onboarding process API requests
      CompanyServlet.java      // Handle company-related API requests
    /models
      User.java                // User entity (company admin)
      Team.java                // Team entity
      Process.java             // Onboarding process entity
      Worker.java              // Worker entity
      Company.java             // Company entity
    /services
      UserService.java         // Logic for user operations
      TeamService.java         // Logic for team operations
      ProcessService.java      // Logic for process operations
      CompanyService.java      // Logic for company operations
    /utils
      DBConnection.java        // Database connection setup
    /config
      DBConfig.java            // DB configurations (JDBC setup)
      AppConfig.java           // Application-level settings
  /frontend
    /assets
      /css
        style.css              // General styles
      /images
      /js
        app.js                  // Frontend JS for AJAX requests
    /views
      index.html               // Home page, welcome/signup/login prompt
      login.html               // Login page
      signup.html              // Signup page
      companyDashboard.html     // Company dashboard page
      teamDashboard.html        // Team dashboard page
      processDashboard.html     // Process dashboard page
      addEditForm.html          // Template for adding/editing forms
    /controllers
      companyController.js      // Frontend logic for company dashboard
      teamController.js         // Frontend logic for team dashboard
      processController.js      // Frontend logic for process dashboard
      formController.js         // Logic for forms (create/edit)
      loginController.js        // Login page frontend logic
```

```
signupController.js    // Signup page frontend logic
common.js              // Shared frontend logic (AJAX, validation)
```

Responsibilities of Each Team Member

1. Yossi: Backend (Java) Responsibilities: - `UserController.java`: Handles HTTP requests related to user operations (login, signup). - `UserService.java`: Business logic for user-related operations (e.g., authentication).

Frontend (JavaScript) Responsibilities: - `loginController.js`: Handles the frontend logic for the login page (sending login request, handling response, etc.). - `index.html`: General home page that welcomes users and redirects to the login or signup page.

2. Idan: Backend (Java) Responsibilities: - `TeamServlet.java`: Handles API requests related to teams (view/edit team data). - `TeamService.java`: Business logic for team operations (add/edit teams).

Frontend (JavaScript) Responsibilities: - `teamController.js`: Handles the frontend logic for the team dashboard page (viewing/editing team information). - `Header&Footer`: General bars for every page.

3. Garmizo: Backend (Java) Responsibilities: - `ProcessServlet.java`: Handles API requests related to onboarding processes (view/edit processes). - `ProcessService.java`: Business logic for process operations (add/edit processes).

Frontend (JavaScript) Responsibilities: - `processController.js`: Handles frontend logic for the process dashboard page (viewing/editing processes). - `login.html`: Page for logging in to the platform.

4. Malki: Backend (Java) Responsibilities: - `CompanyServlet.java`: Handles API requests related to companies (view/edit company details). - `CompanyService.java`: Business logic for company-related operations (add/edit company).

Frontend (JavaScript) Responsibilities: - `companyController.js`: Handles frontend logic for the company dashboard page (viewing company details and summary of processes/teams). - `signup.html`: Page for registering a new company user.

5. Shared: Frontend JavaScript: - `app.js`: Handles global frontend logic such as AJAX functions used across different pages (login, signup, etc.). - `common.js`: Shared functions like form validation and data formatting.

Example of Task Assignment

1. **Yossi:**
 - Backend: Develop `UserServlet.java` for user authentication (login) and `UserService.java`.
 - Frontend: Develop `loginController.js` to handle form submission and communication with the backend.
 2. **Idan:**
 - Backend: Develop `TeamServlet.java` and `TeamService.java` for team management.
 - Frontend: Develop `teamController.js` to manage team-related actions (e.g., viewing/editing teams).
 3. **Garmizo:**
 - Backend: Develop `ProcessServlet.java` and `ProcessService.java` for onboarding process management.
 - Frontend: Develop `processController.js` to manage process-related actions.
 4. **Malki:**
 - Backend: Develop `CompanyServlet.java` and `CompanyService.java` for managing company data.
 - Frontend: Develop `companyController.js` for managing company-related views and actions.
-

Workflow

- Each team member is responsible for both backend (Java) and frontend (JavaScript) logic that deals with their respective area (user, team, process, company).
- The backend (servlets and services) handles API requests, communicates with the database, and sends data to the frontend.
- The frontend controllers handle logic for their respective views, making AJAX requests to the backend and updating the UI based on the responses.
- Common files like `app.js` and `common.js` will be shared between all team members and will be used for global functionalities like AJAX handling, form validation, etc.

This structure allows each member to focus on their area, while also giving them the opportunity to contribute to both backend and frontend development.

Timeline for 24 Hours

Hour 10:00-12:00: Initial Setup

- Set up the project repositories.
- Create the database schema and tables (SQL).

- Set up backend project structure (Java, JDBC).
- Set up the frontend project structure (HTML, CSS, JS).

Hour 12:00-15:00: Frontend Development (Part 1)

- Develop the frontend for the Company Dashboard, Team Dashboard, and Process Dashboard.
- Design the forms for adding/editing teams, processes, and workers.

Hour 15:00-19:00: Backend Development (Part 1)

- Implement Company, User, Team, Process, Stage, Worker Classes.
- Implement the Users Server: Authentication routes (login/signup).
- Implement the Teams Server: CRUD routes for team management.
- Implement the Processes Server: CRUD routes for managing processes.

Hour 19:00-22:00: Backend Development (Part 2)

- Implement the Companies Server: Routes for managing company data.
- Connect JDBC for all servers.
- Begin implementing frontend pages for Company Dashboard, Team Dashboard, Process Dashboard.

Hour 23:00-03:00: Frontend Development (Part 2)

- Finalize all forms: Company user, team creation, onboarding process creation.
- Implement AJAX calls to the REST APIs from frontend pages.

Hour 03:00-05:00: Testing, Debugging & Final Touches

- Test all pages and ensure proper API connections.
- Debug any issues with data flow between frontend and backend.
- Add final CSS styling, ensure responsive design.

Hour 05:00-08:00: SLEEP ?

Hour 05:00-08:00: Presentation Rehearsal