**### \*\*Frontend Pages/Components\*\***

**1. \*\*Login Page UI\*\***

**- \*\*Tasks\*\*:**

**- Design and code the login interface (username/password fields, "Forgot Password" link).**

**- Integrate with backend authentication API and SSO.**

**- Implement form validation and error handling.**

**2. \*\*Event Management UI\*\***

**- \*\*Tasks\*\*:**

**- Code forms for creating/editing events (date picker, location input)(make mobile view and desktop view).**

**- Link to backend `Event Scheduling Logic`.**

**- Add confirmation modals for deletions.**

**3. \*\*Dashboard UI\*\***

**- \*\*Tasks\*\*:**

**- Create a "week-at-a-glance" and monthly view calendar view for events.**

**- Display event cards with location, date, time, and assigned staff.**

**- Add filters/sort options (date, location).**

**4. \*\*User Account UI\*\***

**- \*\*Tasks\*\*:**

**- Design profile management page (edit wage, availability, location).**

**- Connect to backend `User Management API`.**

**- Implement role-based access (Admin vs. Employee views).**

**5. \*\*Scheduling Interface\*\***

**- \*\*Tasks\*\*:**

**- Build drag-and-drop UI for assigning staff to events.**

**- Integrate distance optimization algorithm (auto assign best matches)(recommend binary tree based on wage)a.**

**- Display real-time availability checks.**

**### \*\*Backend APIs/Modules\*\***

**1. \*\*Authentication System\*\***

**- \*\*Tasks\*\*:**

**- Implement JWT token generation/validation.**

**- Secure endpoints (e.g., Admin-only routes).**

**2. \*\*User Management API\*\***

**- \*\*Tasks\*\*:**

**- CRUD operations for employee records (name, wage, availability).**

**- Validate input data (e.g., wage ≥ minimum wage).**

**3. \*\*Event Scheduling Logic\*\***

**- \*\*Tasks\*\*:**

**- Develop algorithm to assign staff based on proximity, availability, and cost.**

**- Calculate travel time/distance using Google Maps API.**

**- Prevent double-booking conflicts per truck.**

**4. \*\*Database API Endpoints\*\***

**- \*\*Tasks\*\*:**

**- Create RESTful APIs for fetching/updating schedules.**

**- Optimize SQL queries for performance (e.g., indexing).**

**---**

**### \*\*System Integration & Testing\*\***

**- \*\*Unit Testing - Frontend\*\*:**

**- Test UI responsiveness and form validations (Jest, React Testing Library).**

**- \*\*Unit Testing - Backend\*\*:**

**- Validate API responses and error handling (Mocha/Chai).**

**- \*\*System Integration Testing\*\*:**

**- Ensure frontend-backend data consistency (e.g., event assignments reflect in the dashboard).**

**---**

**### \*\*Key Dependencies\*\***

**- Frontend pages depend on backend APIs (e.g., Dashboard UI requires `Event Scheduling Logic`).**

**- Authentication must be completed before implementing role-based UIs.**

**- Database schema (PostgreSQL) must align with ERD diagram before API development.**

**Front end**

| **Page/Component** | **Assigned To** | **Key Tasks** |
| --- | --- | --- |
| Login Page UI | Tehjib | Code login form, integrate authentication API, handle error states. |
| Dashboard UI | Chester | Build calendar view, connect to event API, implement filters/sorting. |
| User Account UI | Leona | Develop profile editor, role-based views(emp and admin), link to User Management API. |
| Scheduling Interface | Rudra | Code drag-and-drop assignment UI, integrate distance algorithm, real-time checks. |
| Event Management UI | Kim | Create CRUD forms for events, connect to backend, add confirmation modals. |

**Back end and API**

| **API/Module** | **Assigned To** | **Key Tasks** |
| --- | --- | --- |
| **Authentication System** | **Chester** | **Implement JWT, secure routes, role validation (Admin vs. Employee).** |
| **User Management API** | **Tehjib** | **Build CRUD endpoints for employees, validate wage/availability data.** |
| **Event Scheduling Logic** | **Rudra** | **Develop optimization algorithm, integrate Google Maps API, prevent conflicts.** |
| **Database API Endpoints** | **Kim** | **Design RESTful APIs for schedules, optimize PostgreSQL queries.** |

**Testing and integration**

| **Task** | **Assigned To** | **Key Responsibilities** |
| --- | --- | --- |
| **Unit Testing (Frontend)** | **All** | **Validate UI responsiveness, form submissions, and error handling.** |
| **Unit Testing (Backend)** | **All** | **Test API responses, edge cases, and security checks.** |
| **Integration Testing** | **All** | **Team collaboration to ensure frontend-backend data consistency.** |