**Web Project Documentation**

**Back-end**

**1. Models**

**Creating models:**

* **BusinessAccount Model**: Handles business entities and linked accounts.
* **RolePermission Model**: Manages roles and permissions for multi-user access.
* **Payroll Model**: Tracks payroll cycles, employees, and salary data.
* **Transaction Model**: Manages bulk transactions and their statuses.
* **Invoice Model**: Handles invoice creation and payment statuses.
* **Business Loan Model**: Tracks loan applications and repayment schedules.
* **Analytics Model**: Stores income, expenses, and profitability metrics.
* **Request Model**: Sends requests for changing roles and permissions.

**2. Routes and Controllers**

**Implementing RESTful APIs** for CRUD operations for each feature using Express for routing. Swagger UI is used to test endpoints.

**3. Security using JWT**

* **JWT**: Used for authentication.
* **Role-based access control**: Implemented with middleware.

**Front-end**

**Color Scheme**

* **Primary**: #34343c (dark grey)
* **Accent**:
  + #fc2424 (red)
  + #5373fb (blue)
  + #041c3c (navy blue)
  + #bcf404 (yellow-green)

**1. Pages**

**Components of the front-end** include various pages designed to display and interact with user data.

**2. Connecting API to Components**

Ensure API integration for seamless data flow between the front-end components and the back-end.

**Integrating Features**

**1. Role-specific Access**

Implement role-specific access control to ensure appropriate permissions for different user roles.

**2. Alerts and Notifications**

Integrate real-time alerts and notification features for user interactions and system updates.

**Deployment**