**TOKENTRACKERP PROJECT**

**Introduction**

The TokenTrackProject Management Portal is designed to facilitate project tracking, task allocation, and employee engagement within a company. It includes role-based access control with Superadmin, Admin, and User roles, ensuring a structured and efficient workflow.

**1. Tech Stack**

* **Frontend**: Vue.js
* **Backend**: Laravel API
* **Database**: MySQL with migration files
* **Authentication**: Laravel Sanctum / JWT

**2. Roles & Permissions**

* **Superadmin**:
  + Manages Admin accounts (add, remove)
  + Oversees system settings
* **Admin**:
  + Manages Admin and User accounts
  + Assigns tasks to users
  + Manages project lifecycle
* **User**:
  + Completes assigned tasks
  + Tracks earned tokens
  + Uses Employee Center for requests

**3. Features & Modules**

**3.1 User Management**

* Superadmin creates Admin accounts.
* Admins can add other Admins and Users.
* Users can be deactivated/reactivated.

**3.2 Project Management**

* Create **Projects** with:
  + Internal or External classification
  + Status: **Completed, Incoming, Cancelled, In Progress**
  + Associated Admin and Users and Admin
* Task Management:
  + Assign tasks to Users
  + Track task progress
  + Notify overdue tasks to both User and Admin
  + Token allocation upon task completion

**4.3 Time Tracking**

* Tracks hours spent on each task.
* Tasks nearing the due date trigger notifications.

**4.4 Token & Wallet System**

* Users earn tokens per completed task.
* Admin defines token value (e.g., 1 token = Ksh. 100).
* Wallet tracks token balance & earnings history.

**4.5 Employee Center**

* **Leave Application**: Submit & track leave requests.
* **Complaints**: Report issues with admin review.
* **Refund Requests**: If an employee spends personal funds on company items.

**4.6 Personal Notes**

* Users can save private notes for personal tracking- Can be cleared when browser storage is cleared

**4.7 Dashboard**

* Overview of:
  + Projects: Status tracking (Completed, Incoming, Cancelled, In Progress)
  + Token earnings & balance
  + Task list with deadlines

**5. Database Schema (Migrations)**

* **Users**: id, name, email, role, status, created\_at, updated\_at
* **Projects**: id, name, type (internal/external), status, admin\_id, created\_at
* **Tasks**: id, name, project\_id, assigned\_to, status, due\_date, token\_reward, created\_at
* **Time Tracking**: id, task\_id, user\_id, start\_time, end\_time, duration
* **Tokens**: id, user\_id, task\_id, amount, created\_at
* **Wallet**: id, user\_id, balance, created\_at
* **Employee Requests**: id, user\_id, type (leave, complaint, refund), details, status
* **Notes**: id, user\_id, content, created\_at

**6. API Endpoints**

**Authentication**

* POST /api/login - Authenticate user
* POST /api/register - Register user (Admin only)
* POST /api/logout - Logout user

**Projects**

* POST /api/projects - Create a project
* GET /api/projects - List projects
* PUT /api/projects/{id} - Update project
* DELETE /api/projects/{id} - Delete project

**Tasks**

* POST /api/tasks - Assign task
* GET /api/tasks - List tasks
* PUT /api/tasks/{id} - Update task
* DELETE /api/tasks/{id} - Remove task

**Wallet & Tokens**

* GET /api/wallet - Fetch user wallet
* POST /api/tasks/{id}/complete - Complete task & receive tokens

**Employee Center**

* POST /api/requests - Submit leave/complaint/refund
* GET /api/requests - List requests

**7. Notifications**

* Overdue tasks notify both User and Admin.
* Token earned notifications.
* Admin alerts for pending employee requests.

**8. Deployment & Environment**

* **Local**: Laravel Homestead / Valet, Vue CLI
* **Production**: Nginx, MySQL, Redis for caching

**9. How to Run the Project**

**Backend (Laravel API)**

1. Unzip the file and upload to a Github Repo (Recommended).
2. Navigate to terminal and install dependencies:
3. composer install
4. Create a .env file and configure database settings.
5. Run database migrations:
6. php artisan migrate
7. Start the Laravel server:
8. php artisan serve

**Frontend (Vue.js)**

1. Navigate to the frontend directory and install dependencies:
2. npm install
3. Start the development server:
4. npm run dev

This document provides an overview of the system architecture and feature set. Let me know if further breakdowns or refinements are needed.