## **Project Overview:**

**KQUIRES** is a dynamic and multilingual knowledge base platform. It enables users to efficiently manage articles, categorize content, monitor system statistics, and engage in ticketing and user management. The platform is designed to be easy to use, allowing administrators to interact with the system seamlessly and users to interact with the available features. It supports English and Arabic languages and integrates a robust authentication system.

## **Login Credentials:**

* **UserName:** [superuser@kquires.com](mailto:superuser@kquires.com)
* **Password:** 112233
* **Link:** https://kquires.techanzy.com/

## **Host Configuration:**

* **Host:** KQUIRES
* **HostName:** 40.172.0.91
* **User:** techanzy
* path/.ssh/kquires\_keypair\_2.pem

## **Technologies Used:**

### **Backend:**

* **Django**:
  + A Python-based web framework used for building the backend, handling business logic, database interactions, and serving APIs.
* **PostgreSQL**:
  + A relational database management system used for storing data related to articles, categories, users, and activity logs.

### 

### 

### **Frontend:**

* **Bootstrap**:
  + A popular CSS framework for responsive design, used to style components and ensure the platform works across various screen sizes.
* **CSS**:
  + For custom styling and laying out the platform's interface.
* **i18next**:
  + A JavaScript library used for internationalization (i18n) and localization, allowing the platform to support multiple languages (English and Arabic).
* **JavaScript**:
  + Used for dynamic behavior, API calls, and front-end interactivity.

### **Authentication & Authorization:**

* **Django Allauth**:
  + Handles user authentication, including registration, login, and password management.

### **Miscellaneous Libraries:**

* **CKEditor**:
  + A rich text editor used for creating and editing content in articles, providing a WYSIWYG (What You See Is What You Get) experience for content creators.

### **Web Hosting/Deployment:**

* **AWS**:
  + For deploying the backend and frontend services.
* **Docker**:
  + For containerizing the application, making it easier to deploy on different environments.

## **Key Features:**

### **Dashboard:**

* Displays an overview of total articles, pending articles, approved articles, and rejected articles.
* Users can create new articles and categories directly from the dashboard.
* Displays quick statistics and action buttons for easy access to articles and categories.

### **Articles:**

* Manage articles with detailed information like title, publication date, status (approved/rejected/pending), and comments.
* Articles can be edited or deleted, and approval/rejection can be done directly through the dashboard.

### **Categories:**

* Organize articles into categories and subcategories.
* Ability to add new categories, edit existing ones, and manage their status.
* Categories are displayed with the number of articles in each.

### **User Management:**

* Manage users, including their permissions, status, and department.
* Admins can edit user details or deactivate users as needed.

### **Ticket Management:**

* Create, manage, and track the status of tickets (solved or pending).
* Tickets contain a unique ticket number, title, status, and preview.
* Users can edit or delete tickets, and administrators can approve/reject them.

### **Notification Messages:**

* Send notification messages to users and control their visibility.
* Each message can be edited or deleted, and it shows the publication date and author.

### **Activity Logs:**

* Keep track of all user activities in the system such as login, logout, and updates on articles or tickets.
* Each activity is logged with the date/time, username, and a description of the action performed.

### **Multilingual Support:**

* The platform supports English and Arabic languages.
* Users can switch between languages using a language toggle in the top navigation.

### **Sign-Out Confirmation:**

* A confirmation dialog appears when users attempt to sign out, ensuring that sign-out actions are intentional.

## **Users Roles and Permissions**

### **1. Visitor:**

* **Access:**
  + Can view article content only. No actions (editing, deleting, or submitting).
* **Displayed Information:**
  + No details such as date published, creator name, or metadata.

### **2. Agent/Employee:**

* **Authentication:**
  + Sign in / Sign up available.
* **Access:**
  + KBS Homepage access
  + Search articles
  + Change password
  + Submit tickets

### **3. Content Manager:**

* **Authentication:**
  + Sign in / Sign up available.
* **Access:**
  + KBS Homepage access
  + Search articles
  + Change password
  + Submit tickets
* **Content Management:**
  + Create/update articles (no delete)
  + Create/update categories (no delete)
  + Multi-department selection for categories
* **Dashboard:**
  + View article statistics (Pending, Approved, Published, Total)
  + Quick action buttons: New Article, New Category, Approval Manager

### **4. Approval Manager:**

* **Authentication:**

* + Sign in / Sign up available.
* **Access:**
  + KBS Homepage access
  + Search articles
  + Change password
  + Submit tickets

### **5. Article Management:**

* + Approve/reject articles
  + Add comments
  + Edit and delete articles and categories
* **Dashboard:**
  + View article statistics
  + Manage articles and users
  + Access Statistics Page

### **6. Admin:**

* **Full Permissions:**
  + Complete access to all system functions.
* **Department Management:**
  + Create/edit/delete departments (no duplicates)
  + Manage departments via Admin panel
* **User Management:**
  + Assign departments to users
* **Article and Category Management:**
  + Approve/reject, edit, delete articles and categories
* **Full System Access:**
  + Full rights to all sections, including Dashboard and Statistics Page

## **Conclusion:**

This documentation provides an overview of the **KQUIRES**, including its technologies, features, user roles, and permissions. The platform is designed for easy content management and provides powerful tools for admins, content managers, and agents to collaborate and maintain the system.