Изображение выглядит как текст, диаграмма, Параллельный, линия

Контент, сгенерированный ИИ, может содержать ошибки.

# e-Tendering System Functional Documentation

Technology Stack

React, Node.js, Express, MongoDB

Project Structure

├── index.html

├── server.js

└── src/

├── App.css

├── App.jsx

├── index.css

├── main.jsx

├── utils/

├── validation.js

├── components/

├── citizenPage

│ ├── CitizenFeedback.jsx

│ ├── CitizenPage.jsx

│ ├── ClosedTenders.jsx

│ ├── OpenTenders.jsx

│ ├── ViewFeedback.jsx

├── cityPage

│ ├── CategoryDashboard.jsx

│ ├── ChoiceDashboard.jsx

│ ├── CityPage.jsx

│ ├── ManageCategories.jsx

│ ├── ManageUserCategories.jsx

│ ├── ManageUsers.jsx

│ ├── ViewFeedbacks.jsx

├── companyPage

│ ├── CompanyApplication.jsx

│ ├── CompanyBids.jsx

├── CreateTender.jsx

├── CreateUser/

│ ├── CreateTender.jsx

│ ├── CreateAdmin.jsx

├── DetailedInfo/

│ ├── DetailedInfo.jsx

│ ├── DetailedBids.jsx

├── Home.jsx

├── Login.jsx

├── Navbar.jsx

├── SubmitBid.jsx

├── TenderList.jsx

├── UpdateTender.jsx

├── UpdateUser.jsx

Running the Project

* **Prerequisites**
  + Node.js (v20.x)
  + MongoDB Atlas or a local MongoDB instance

**Setup**

**1. Clone the repository:**

git clone https://e2205685@dev.azure.com/e2205685/e-Tendering/\_git/e-Tendering

cd e-Tendering

**2. Install dependencies:**

npm install

**3. Start the server:**

npm start

**4. Start the front-end development server:**

npm run dev

Main Features

citizenPage/

* **CitizenFeedbacks.jsx:** Allows citizens to submit feedback on tenders and view feedback
* **CitizenPage.jsx:** Main page for citizens to choose between open and closed tenders
* **OpenTenders.jsx:** List tenders that are currently open for bidding
* **ClosedTenders.jsx:** Displays tenders that have been closed and are no longer accepting bids.
* **ViewFeedback.jsx:** Enables users to browse on tenders

cityPage/

* **CategoryDashboard.jsx:** Display a dashboard for managing tender categories
* **ChoiceDashboard.jsx:** Selection interface for choosing bid winner
* **CityPage.jsx:** The main city administration page
* **ManageCategories.jsx:** Allows city officials to create, edit, and delete tender categories.
* **ManageUsers.jsx:** Managing user accounts.
* **ViewFeedbacks.jsx:** Display all user feedback for all tenders.

companyPage/

* **CompanyApplication.jsx:** Application documents for the Tenders
* **CompanyBids.jsx:** Displays all bids submitted by the logged-in company

Tender Management/

* **CreateTender.jsx:** A form for creating new tenders.
* **TenderList.jsx:** Displays a list of available tenders.
* **UpdateTender.jsx:** Allows administrators to update tender details.
* **SubmitBid.jsx:** Interface for companies submitting bids amount

User Management/

* **CreateUser.jsx:** A form for creating new users.
* **CreateAdmin.jsx:** A form for creating new users (as a city staff).
* **UpdateUser.jsx:** Allows administrator to update user information

Detailed Info/

* **DetailedInfo.jsx:** Shows all information about tenders (ID, Descrition, ...)
* **DetailedBids.jsx:** Displays all bids for a specific tender

Other Components/

* **Home.jsx:** Home page of the application
* **Login.jsx:** User authentication and login page
* **Navbar.jsx:** The navigation bar for switching different setions of the app.

API Endpoints

|  |  |  |
| --- | --- | --- |
| Endpoint | Method | Description |
| /save\_tender | POST | Creates a new tender. |
| /find | GET | Fetches all tenders with populated bids and winner data. |
| /create\_user | POST | Registers new user. |
| /users | GET | Fetches all users with populated categories and user types. |
| /users/:user\_id | GET | Fetches a user by their user\_id. |
| /user\_types | GET | Fetches all user types. |
| /create\_user\_type | POST | Creates a new user type. |
| /delete\_user/:user\_id | DELETE | Deletes a user by their user\_id. |
| /delete\_tender/:tender\_id | DELETE | Delete a tender by its tender\_id |
| /delete\_bid/:bid\_id | DELETE | Deletes a bid by its bid\_id |
| /delete\_category/:category\_id | DELETE | Deletes a category by its category\_id |
| /create\_bid | POST | Creates a bid linking a user and a tender. |
| /bids | GET | Fetches all bids with populated user and tender data |
| /create\_category | POST | Creates a new category |
| /categories | GET | Fetches all categories with populated user data. |
| /add\_user\_to\_category | POST | Adds a user to multiple categories. |
| /add\_user\_to\_specific\_category | POST | Adds a user to a specific category by their IDs. |
| /remove\_user\_from\_category | POST | Removes a user from a specifc category. |
| /update\_tender/:tender\_id | PUT | Updates a tender's winner and status |
| /update\_user/:user\_id | PUT | Updates user details by user\_id |
| /update\_bid/:bid\_id | PUT | Updates bid details by bid\_id |
| /users\_categories | GET | Fetches all user-category relationships with populated user and category data. |
| /feedback | GET | Fetches feedback for a specific user |
| /submit\_feedback | POST | Submits feedback for a user or a tender |

Role-Based Access Control

|  |  |
| --- | --- |
| Role | Permissions |
| Citizen | View tenders, submit feedback, view bids |
| Company | Submit bids, view own bids, view tenders |
| City Staff (Admin) | Full access to all features |