**Email Address Tracking Application Documentation**

**Table of Contents**

* [Introduction](https://chat.openai.com/?model=text-davinci-002-render-sha#introduction)
* [Installation](https://chat.openai.com/?model=text-davinci-002-render-sha#installation)
* [Usage](https://chat.openai.com/?model=text-davinci-002-render-sha#usage)
* [Configuration](https://chat.openai.com/?model=text-davinci-002-render-sha#configuration)
* [Routes](https://chat.openai.com/?model=text-davinci-002-render-sha#routes)
* [Controllers](https://chat.openai.com/?model=text-davinci-002-render-sha#controllers)
* [Models](https://chat.openai.com/?model=text-davinci-002-render-sha#models)
* [Middleware](https://chat.openai.com/?model=text-davinci-002-render-sha#middleware)
* [Error Handling](https://chat.openai.com/?model=text-davinci-002-render-sha#error-handling)
* [Testing](https://chat.openai.com/?model=text-davinci-002-render-sha#testing)
* [Deployment](https://chat.openai.com/?model=text-davinci-002-render-sha#deployment)
* [Contributing](https://chat.openai.com/?model=text-davinci-002-render-sha#contributing)
* [License](https://chat.openai.com/?model=text-davinci-002-render-sha#license)

**Introduction**

The Email Address Tracking Application is designed to help the Taita Taveta county government manage their staff’s email addresses effectively. It provides functionality for users to recover their forgotten email addresses, request password resets, and request new email addresses. The application includes two types of admins: Super Admin and Normal Admin. The Super Admin has additional privileges, such as managing other admins.

**Installation**

1. Clone the repository from GitHub.
2. Install the required dependencies using ’npm install’.
3. Set up the database and configure the connection.

* Create an empty mysql database
* Import existing database to avoid data loss
* Create database user and password

1. Configure the application settings, including the email portal credentials by adding the following to the environmental variables.

* **WhatsApp Credentials (** *for sending WhatsApp messages to the selected Admin***)**
* API\_TOKEN\_INSTANCE – obtained from greenapi.com
* ID\_INSTANCE – obtained from greenapi.com
* ADMIN1 – phone number to receive WhatsApp texts.
* **Database Credentials**
* DB\_DIALECT - mysql
* DB\_HOST - mcrh.or.ke
* DB\_NAME - database name you created above
* DB\_USER – user created
* DB\_PASS – database user password
* **Mail User Credentials (** *used as the primary email for sending requests to admins***)**
* MAIL\_HOST - mail.taitataveta.co.ke
* MAIL\_USER - [ictsupport@taitataveta.co.ke](mailto:ictsupport@taitataveta.co.ke)
* MAIL\_PASS – password for the email above
* **SMS Credentials (** *used for sending feedback to users***)**
* SMS\_API\_KEY - d2c8bdf6c1d5576ae0c12c9785dfb5e59c7e4717701c4ed9c9c6acfb50a85900

**N/B**: All environmental variables are case sensitive, they should be declared the as stated above to avoid any mistakes.

1. Run the application using npm start.

**Usage**

**User Perspective:**

**Confirm your email address and reset password**

1. Visit the application's homepage.
2. Provide your first and last name in the input fields.
3. Click the "Submit" button.
4. If your email address exists in the database, you will be redirected to a results page displaying your full name, correct email address, and a field to update your phone number for optional feedback purposes.
5. Click the "Submit" button on the results page to create a password reset request.
6. You will receive an SMS notification with the phone number you provided to confirm that your request has been received by the admins and that you will be notified once they are done resetting your email.
7. The admin will handle your request and reset your password.
8. You will receive an SMS notification informing you that your email address has been reset successfully.

**Requesting new email address**

1. Visit the application's homepage.
2. Provide your first and last name in the input fields.
3. Click the "Submit" button.
4. Since your email address does not exist in the database, you will be redirected to a results page displaying your full name, and a field to update your phone number for optional feedback purposes.
5. A retry button is also helpful to confirm that you did not make any mistakes while typing your name.
6. Click the "Submit" button on the results page to request a new email address.
7. You will receive an SMS notification with the phone number you provided to confirm that your request has been received by the admins and that you will be notified once they are done creating your email.
8. The admin will handle your request and create your new email address.
9. You will receive an SMS notification informing you that your email address has been created.

**Admin Perspective**

1. **Creating admin account.**

There are two ways to create admin account:-

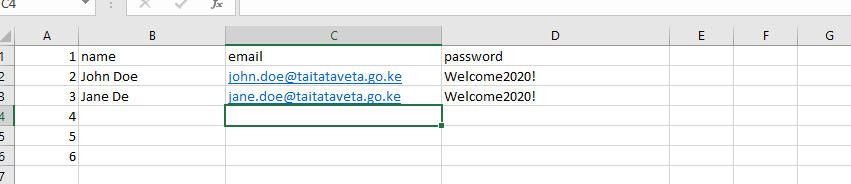
1. Sign up via the app and wait for super Admin to activate your account before you can login to the system.
2. Super Admin on their end are able to create you a new account.

Once your account has been activated you will now start receiving emails with requests from users proceed to login and perform actions on the dashboard.

1. **Handling User Requests.**
2. Log in to the application dashboard using your official email address.
3. View the requests submitted made by users listed on the first page.
4. Log in to the zimbra email portal and create/reset the user's email.
5. Return to the application's dashboard to complete the requests.
6. The user will receive a notification about the successful password reset.
7. View your recent handled requests actions from the recent reset requests tab.
8. **Importing Newly created emails.**

The system allows the admins to add newly created mails in bulk via the import tab in the dashboard. Before importing the mails it should be noted that this version of the system only allows imports in form of CSV(comma separated values) files, after generating the excel from the Zimbra portal the admins are required to do the following.

1. Open the excel file with MS Excel
2. The file should be formatted to match the following screenshot.



Note that the field names should be in lowercase and always the first row

1. Make sure to save the file as CSV file e.g. newMails.csv.
2. Head over to the import tab on the dashboard.
3. Locate the file you just created and click import.
4. The system will process the request and return the newly created mails listing in descending order of how they were created.
5. You have successfully imported the mails into the system. Users will now be able to view their mails from the user’s perspective.

**Routes**

* **POST /users/recover**: Checks if the user's email exists in the database based on first and last name. If found, redirects to the results page.
* **POST /users/password-reset**: Creates a password reset request for an admin to handle.
* **POST /users/email-request**: Creates an email creation request for an admin to handle.

**Controllers**

* **UserController**: Handles user-related actions, such as email recovery, password reset requests, and email creation requests.

**Models**

* **User**: Represents a user in the system, including attributes such as first name, last name, email address, and phone number.
* **PasswordResetRequest**: Represents a password reset request made by a user, including attributes such as user ID and status.
* **EmailCreationRequest**: Represents an email creation request made by a user, including attributes such as user ID and status.

**Middleware**

* **AuthenticationMiddleware**: Validates user authentication and authorization.
* **ErrorHandlingMiddleware**: Handles errors and sends appropriate responses.
* **ValidationMiddleware**: Validates user input and request data.

**Error Handling**

The application uses HTTP status codes and custom error messages to handle errors. Possible error scenarios include invalid input, unauthorized access, and database errors.

**Testing**

The application uses the Mocha testing framework. To run the tests, use the command npm test. Write tests for the controllers, models, and any custom middleware to ensure the application functions as expected.

**Deployment**

To deploy the application to a production environment, follow these steps:

1. Set up a hosting server or cloud service provider.
2. Configure the environment variables for the database connection, email portal credentials, and other settings.
3. Set up a reverse proxy or load balancer if required.
4. Deploy the application using a process manager like PM2 or as a Docker container.

**Contributing**

Contributions to the Email Address Tracking Application are welcome. To contribute, follow these steps:

1. Fork the repository on GitHub.
2. Set up a development environment using the installation instructions.
3. Make your changes and test them thoroughly.
4. Submit a pull request with a clear description of your changes and their benefits.

**License**

The Email Address Tracking Application is released under the [MIT License](https://opensource.org/licenses/MIT). You are free to use, modify, and distribute the application as per the terms of the license.

Please note that this is just a template, and you should customize it further based on your specific implementation, code structure, and requirements.