API Documentation

Base URL

`http://localhost:3000`

User APIs

1. Signup

Endpoint:

`POST /api/users/signup`

Description: Registers a new user.

Request Body:

```json
{
  "username": "string",
  "email": "string",
  "password": "string",
  "phone": "string"
}
```

Response: - **201 Created** json { "message": "User registered successfully", "user": { "username": "string", "email": "string" }, "accessToken": "jwt-access-token", "refreshToken": "jwt-refresh-token" } - **400 Bad Request** json { "error": "All fields are required" } - **400 Bad Request** json { "error": "Email already exists" } - **500 Internal Server Error** json { "error": "Internal server error" }

---

2. Login

Endpoint:

`POST /api/users/login`

Description: Logs in a user with valid credentials.

Request Body:

```json
{
  "email": "string",
  "password": "string"
}
```

Response: - **200 OK** json { "message": "Login successful", "user": { "username": "string", "email": "string" }, "accessToken": "jwt-access-token", "refreshToken": "jwt-refresh-token" }
- **400 Bad Request** json { "error": "Email and password are required" } - **401 Unauthorized** json { "error": "Invalid credentials" } - **500 Internal Server Error** json { "error": "Internal server error" }

---

3. Refresh Token

Endpoint:

`POST /refresh`

Description: Exchanges a valid refresh token for a new access token and refresh token.

Request Body:

```json
{
  "refreshToken": "jwt-refresh-token"
}
```

Response: - **200 OK** json { "accessToken": "new-jwt-access-token", "refreshToken": "new-jwt-refresh-token" } - **400 Bad Request** json { "error": "Refresh token required" } - **401 Unauthorized** json { "error": "Refresh token expired" } - **401 Unauthorized** json { "error": "Invalid refresh token" }

Contact Us (Ticket Creation)

Endpoint:

`POST /contact-us`

Description: Submits a contact request, stores it as a ticket in the database (with status 'open'), and sends an email to all admin emails.

Request Body:

```json
{
  "name": "string",
  "email": "string",
  "message": "string"
}
```

Response: - **200 OK** json { "success": true, "message": "Message sent successfully.", "sentTo": [{ "email":

```
"admin1@example.com" },        { "email": "admin2@example.com"
}    ],     "ticket": {       "_id": "string",        "name":
"string",        "email": "string",        "message": "string",
"status": "open",       "createdAt": "2025-07-12T03:27:14.100Z",
"updatedAt": "2025-07-12T03:27:14.100Z"     }   }
```

- **400 Bad Request**
```
json {     "error": "Name, email, and message are
required."   }
```
- **500 Internal Server Error**
```
json   {      "error":
"Failed to save ticket."   }
```
or
```
json   {      "error": "Failed to
fetch admin emails."   }
```
or
```
json   {      "error": "Failed to send
message."   }
```

Notes: - All errors are logged to `logs/error.log`. - Tickets are created with status `open` by default. - Admin emails are fetched from the database.