Contents

[About The Project 2](#_Toc72716255)

[Dependencies 2](#_Toc72716256)

[Architecture 2](#_Toc72716257)

[Getting Started 2](#_Toc72716258)

[Prerequisites 2](#_Toc72716259)

[Reactjs Frontend 3](#_Toc72716260)

[Installation 3](#_Toc72716261)

[Installation with docker: 3](#_Toc72716262)

[Nodejs backend 3](#_Toc72716263)

[Installation 3](#_Toc72716264)

[Using docker-compose: 4](#_Toc72716265)

[Usage 4](#_Toc72716266)

[Credentials for logging in 4](#_Toc72716267)

[Application Screenshots 5](#_Toc72716268)

[Testing Screen Shot 6](#_Toc72716269)

[Request/ Response 7](#_Toc72716270)

[**Contributing** 8](#_Toc72716271)

## About The Project

Build an incident management service should have the following features:

* Raise an incident as an admin
* Assign the incident to a user
* Acknowledge the incident as a user
* Resolve the incident as a user
* Read details about a certain incident
* Index incidents (includes filtering, sorting by date created/updated and incident type and paging)
* Delete an incident

## Dependencies

* Reactjs- Redux
* nodejs
* mongodb [Note : This is running in cluster mode under xxxx]

## Architecture

Diagram

Description automatically generated

## Getting Started

### Prerequisites

This is an example of how to list things you need to use the software and how to install them.

Node: npm install node@latest -g or download LTS version from node site

## Reactjs Frontend

### Installation

Clone the repo

git clone <https://github.com/Poonam1910/Ticket-Management-System.git>

Install NPM packages

npm install

Build the project:

npm run build

Run the cypress test:

npm test

Run the project:

npm start

### Installation with docker:

Clone the repo:

git clone <https://github.com/Poonam1910/Ticket-Management-System.git>

Build the docker image:

docker build -t <image\_name>/<image\_tag> -f Dockerfile.prod .

Run the docker image:

docker run --name <container\_name> -p 80:80 -d <image\_name>/<image\_tag>

## Nodejs backend

### Installation

Clone the repo

git clone <https://github.com/Poonam1910/Ticket-Management-Server-API.git>

Install NPM packages

npm install

Build the project:

npm run build

Run the jest test:

npm test

Run the project:

npm run app

Installation with docker:

Clone the repo:

git clone <https://github.com/Poonam1910/Ticket-Management-Server-API.git>

Build the docker image:

docker build -t <image\_name>/<image\_tag> -f Dockerfile.prod .

Run the docker image:

docker run --name <container\_name> -p 8082:8082 -d <image\_name>/<image\_tag>

### Using docker-compose:

Start the frontend and backend services:

docker-compose -f <docker-compose.yml> up -d

Stop the frontend and backend services:

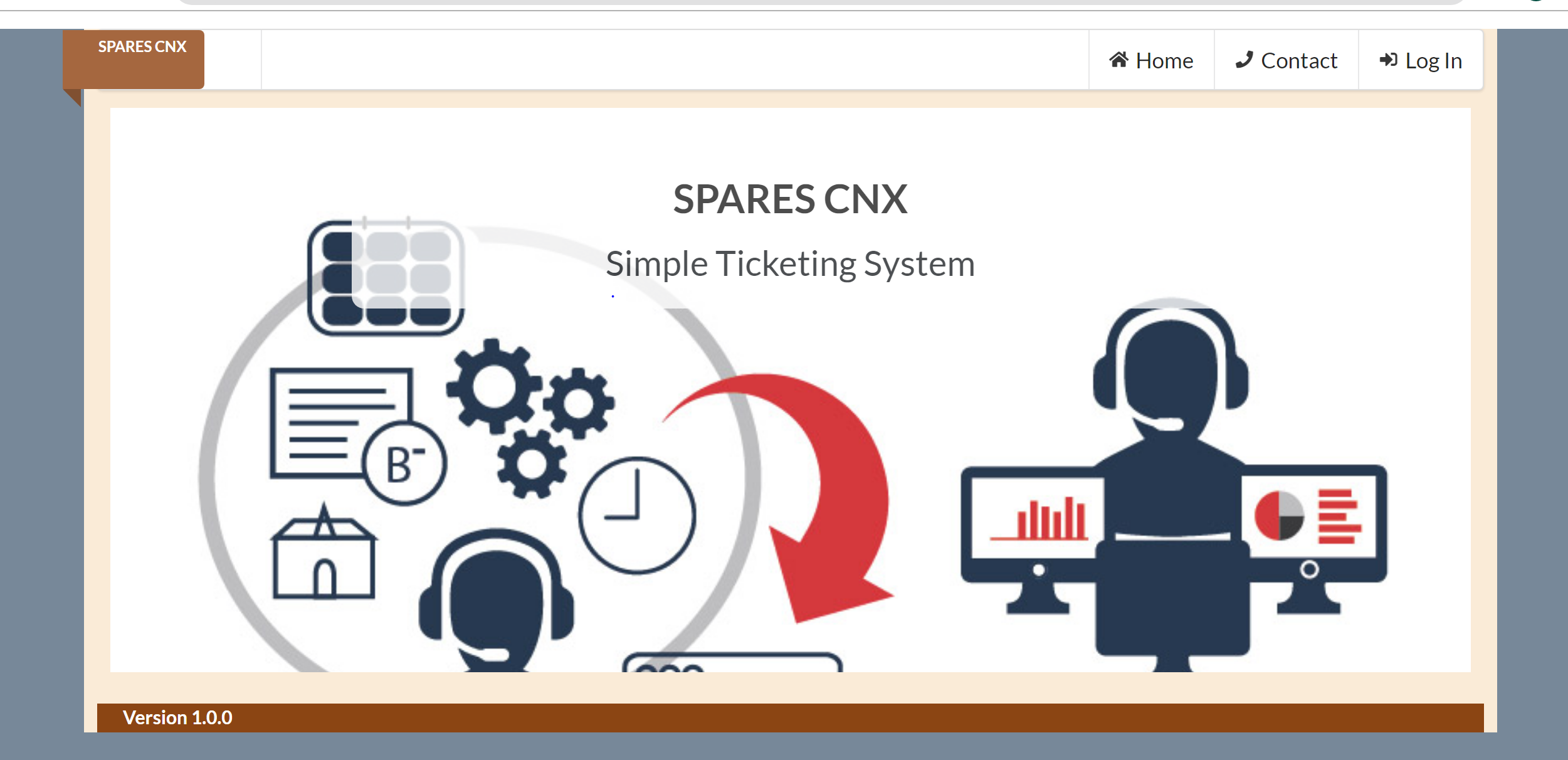
docker-compose -f < docker-compose.yml> down

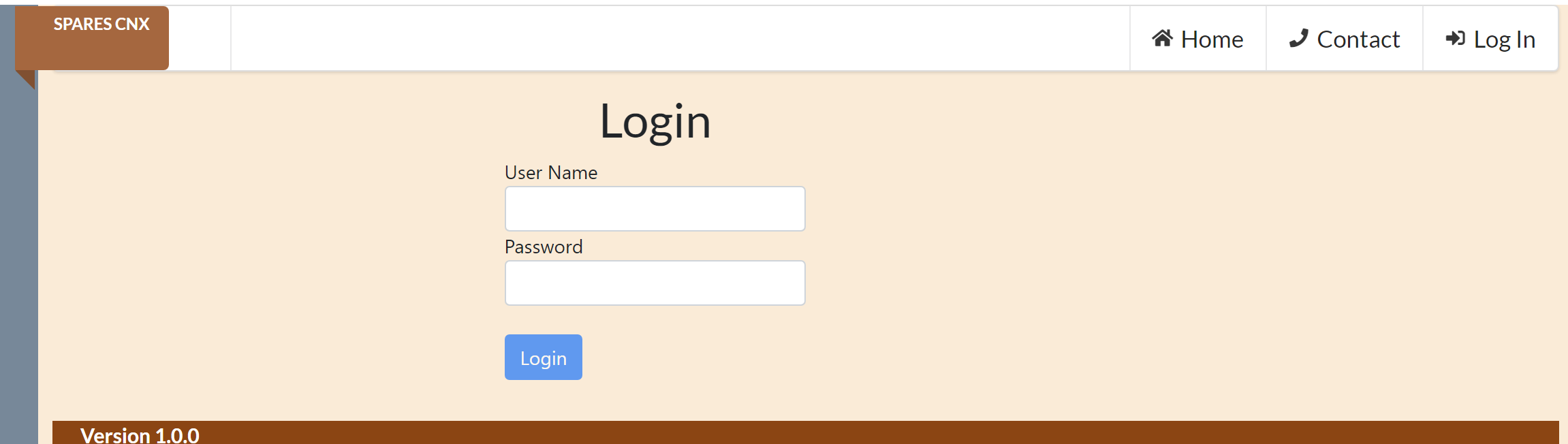
## Usage

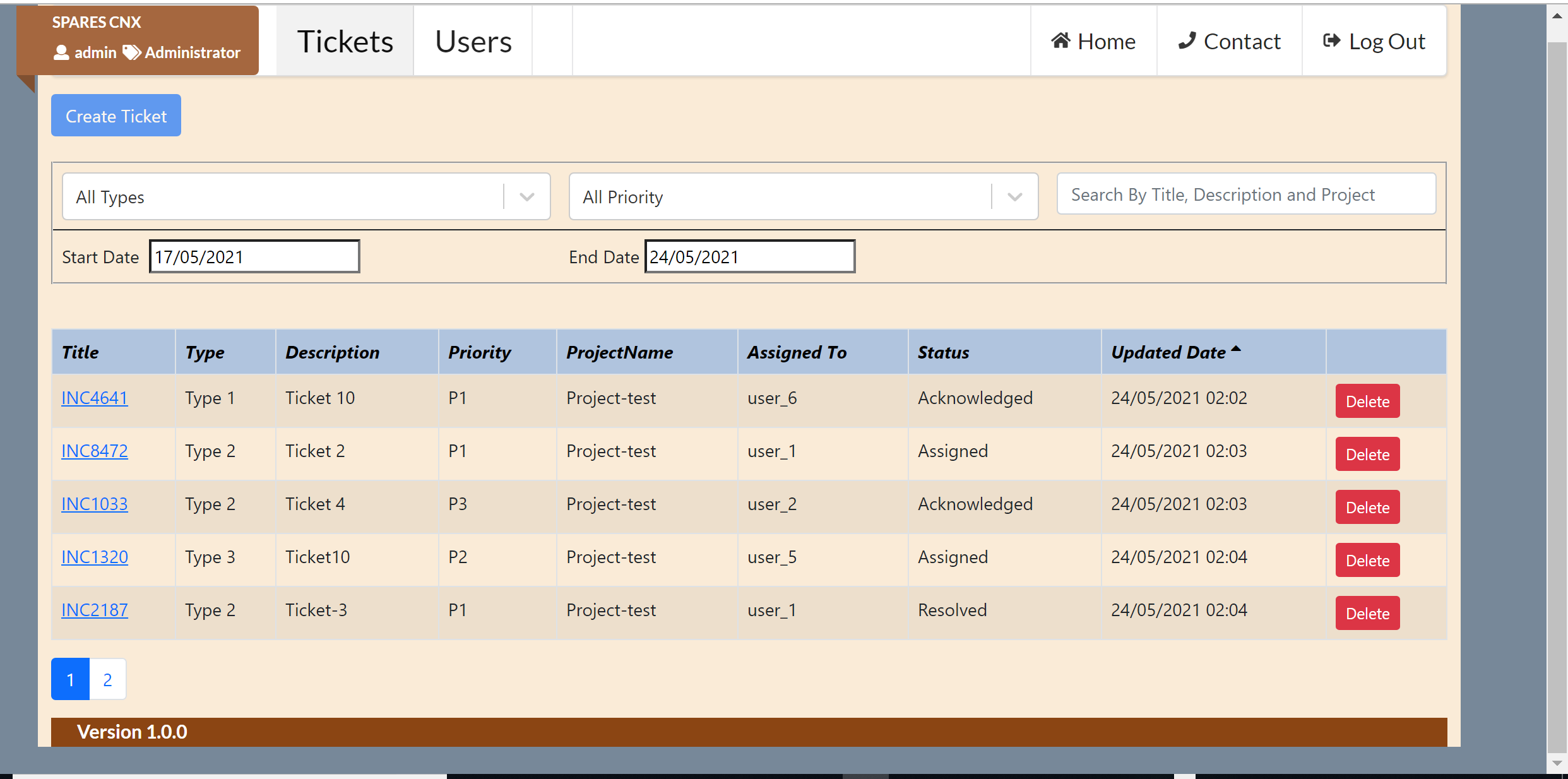
### Credentials for logging in

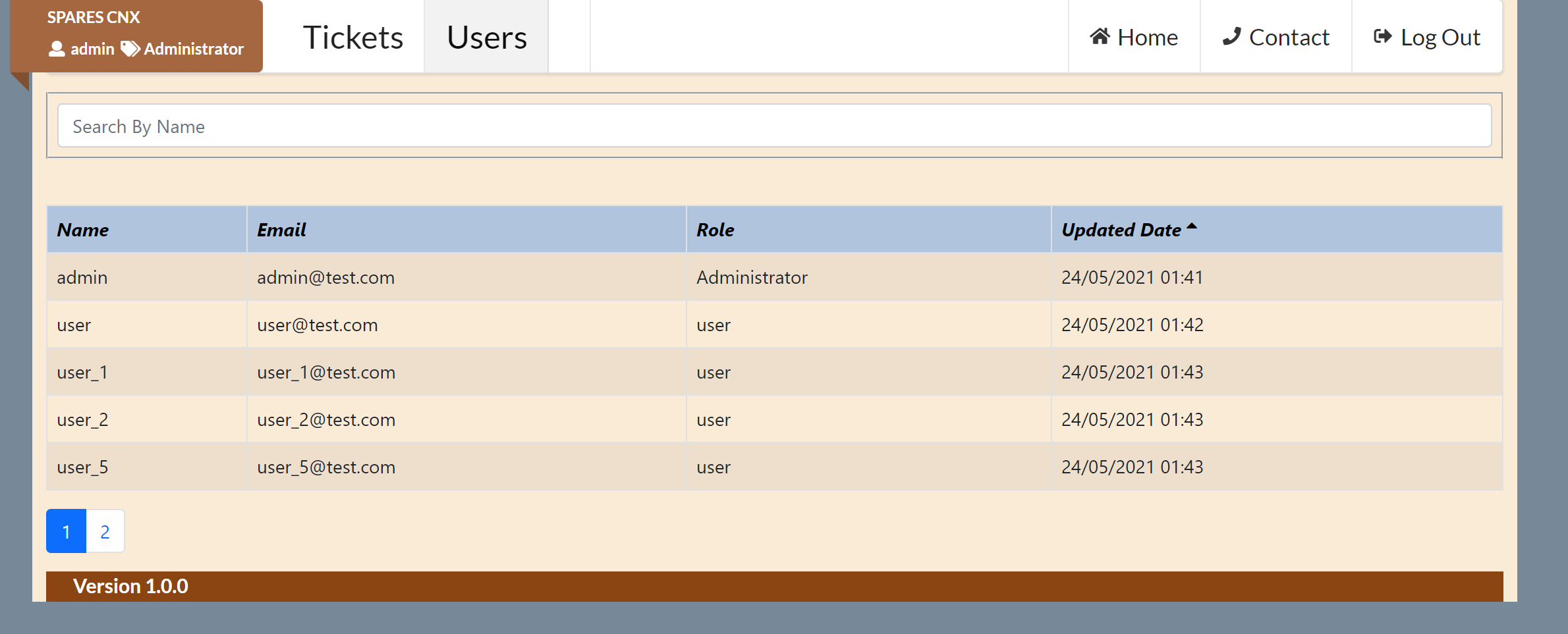
|  |  |  |
| --- | --- | --- |
| UserType | Username | Password |
| Admin | Admin | admin |
| User | User | user |

## Application Screenshots

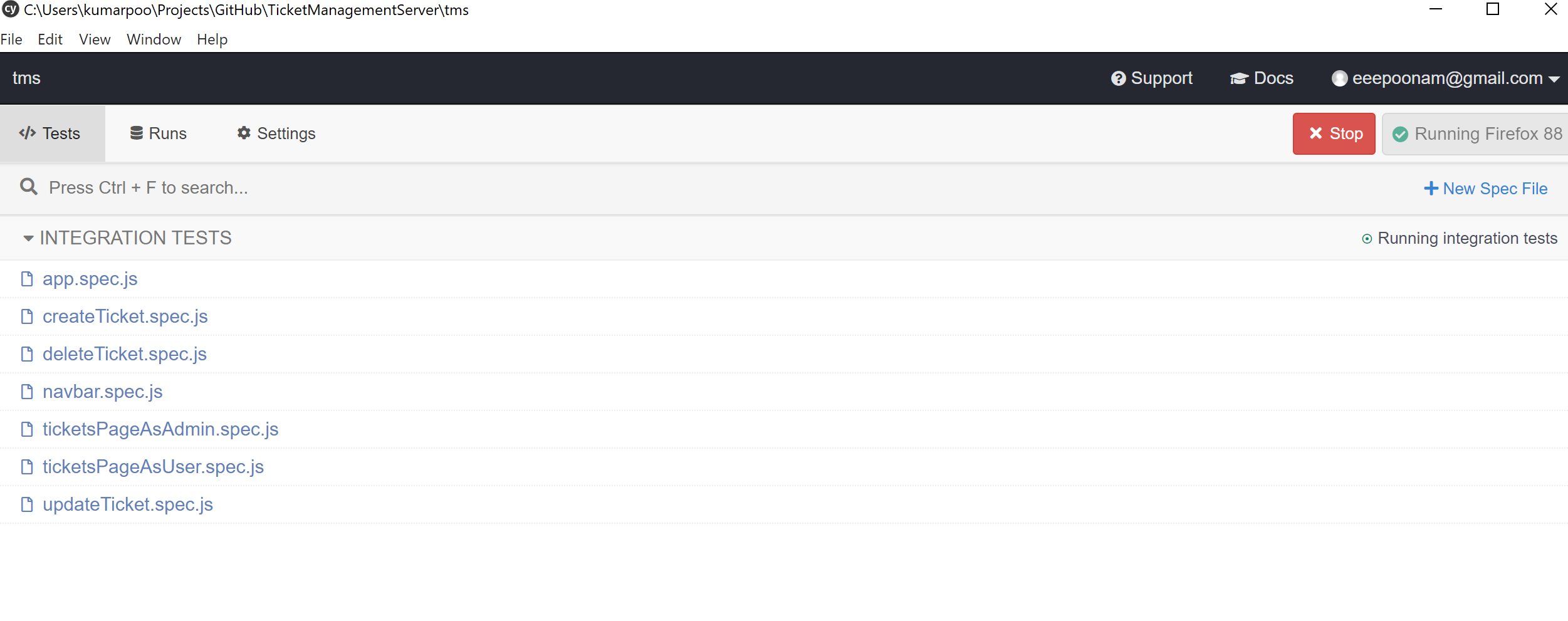


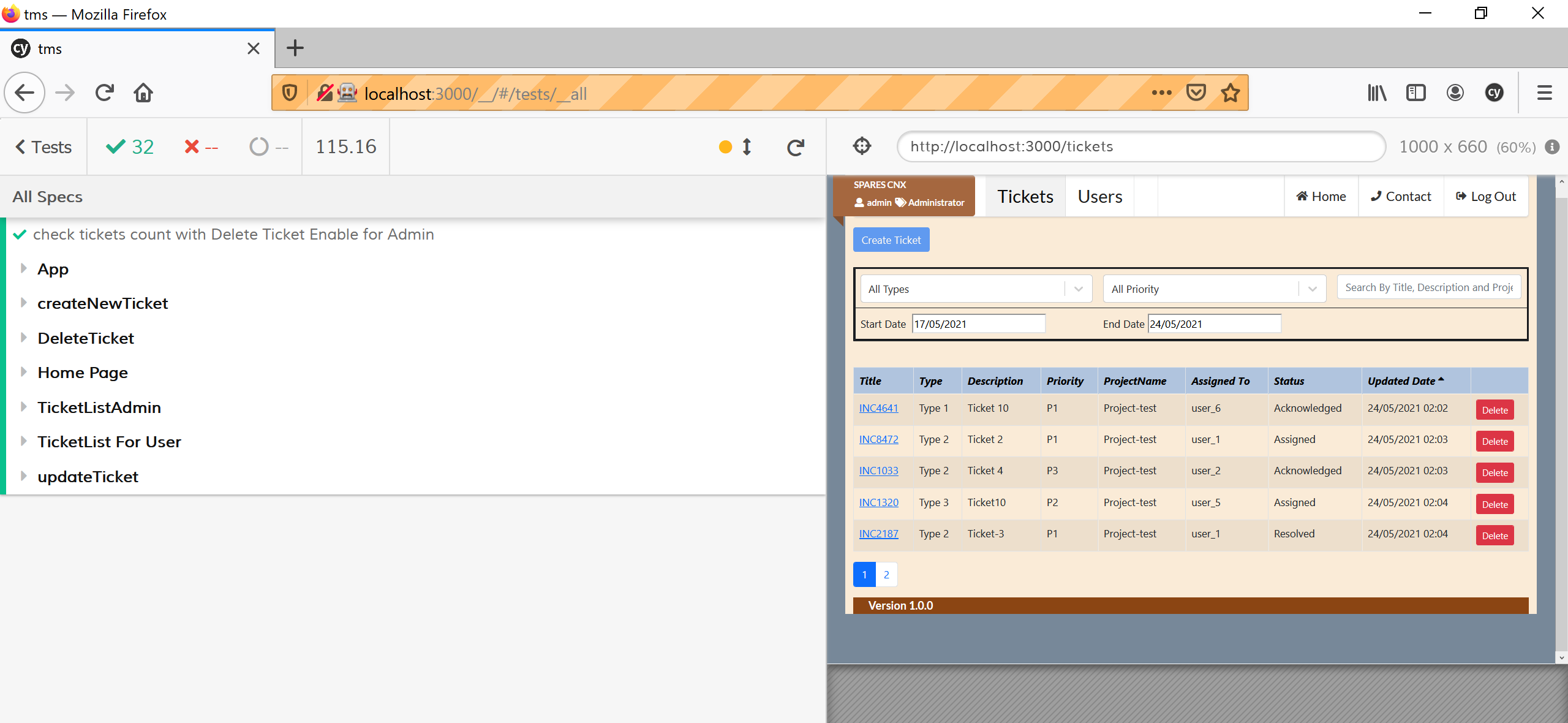






## Testing Screen Shot





## Request/ Response

CRUD operation is available for Tickets through UI , Similar can be replicated for User.

USER/TICKET CRUD API are available in node project.

Sample json request to create new user

Default port exposed for API: 8082 and for UI as 3000.

Please change the port & host based in application config files based on your exposed Ep.

User Create: http://localhost:XXXX/users/create

User

Get: http://localhost:XXXX/users

Post: <http://localhost:XXXX/users/create>

Put: http://localhost:XXXX/users/<Id>

Delete: http://localhost:XXXX/users/XXX

Json:

{

"name":"user\_6",

"role":"user",

"email":"user\_6@test.com",

"loginName":"user\_6"

}

Tickets

Get: http://localhost:XXXX/tickets

Post: <http://localhost:XXXX/tickets/create>

Put: http://localhost:XXXX/tickets/<Id>

Delete: <http://localhost:XXXX/tickets/XXX>

Json:

{"description": "tes", "projectName": "Project-test", "assigneeId": "60a0c8c9eb957939600487a6",

"priorityId": 1, "typeId": 1,"statusId":1}

### **Contributing**

Contributions are what make the open source community such an amazing place to be learn, inspire, and create. Any contributions you make are **greatly appreciated**.

1. Fork the Project
2. Create your Feature Branch (git checkout -b feature/AmazingFeature)
3. Commit your Changes (git commit -m 'Add some AmazingFeature')
4. Push to the Branch (git push origin feature/AmazingFeature)
5. Open a Pull Request