

## Overview

HH is a project to manage patient transfers between hospitals and clinics. Its features allow the user to login with proper authentication. The user would have access to the information of a module Patients allowing them to view and modify the database.

## Technologies used:

[Node.js](#)- This is a cross-platform runtime environment built on Chrome's V8 JavaScript engine used in running JavaScript codes on the server. It allows for installation and managing of dependencies and communication with databases.

[ExpressJS](#)- This is a NodeJS web application framework.

[Mysql](#)- Open-source relational database management system.

## Database setup

### 1.Install mysql client

[MySQL :: Getting Started with MySQL](#)

### 2.Create new user

**sudo mysql -u root -p**

Launch the MySQL shell as the root use

**CREATE USER 'username' IDENTIFIED BY 'password';** Create user by replacing username and password with username and password of your choice.

**CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';** If you are working locally on the machine with MySQL, use username@localhost to define the user.

**CREATE USER 'username'@'ip\_address' IDENTIFIED BY 'password';** If you are connecting remotely, use username@ip\_address, and replace ip\_address with the actual address of the remote system hosting MySQL.

### 3.Create database

**CREATE DATABASE database\_name;**

Example: **CREATE DATABASE hh\_backend;**

### 4. Import sql file

**mysql -u username -p database\_name < file.sql**

Example: **Mysql -u root hh\_backend <~/hh\_backend/Project/hh\_backend.sql**

## Setup:

Get the code [url:HH Backend](#)

```
npm install //to install all dependencies
```

Create an .env file in your project root folder and add your variables. Example of DotEnv.

### .env

APP\_PORT= PORT #

DB\_PORT= DATABASE PORT #

DB\_HOST= DATABASE HOST

DB\_USER= USER

DB\_PASS= PASSWORD

DB\_NAME=hh\_backend

SECRET=

JWT\_SECRET =

JWT\_EXPIRES\_IN = 90d

JWT\_COOKIE\_EXPIRES = 90

## Start server:

```
npm start
```

## API Endpoints

HTTP Verbs	Endpoints	Action
------------	-----------	--------

POST	/api/login	To login an existing user account
GET	/api/login	To log out user

POST	/api/users	To create user
GET	/api/users	To find user
GET	/api/users/id	To find user by id
PUT	/api/users/id	To update user by id
DELETE	/api/users/id	To delete user by id

GET	/api/patients	Show patients
POST	/api/patients	Create patients
GET	/api/patients/id	show patient by id
DELETE	/api/patients/id	to delete patient by id
PATCH	/api/patient/id	to update patient by id

POST	/api/transfers	Create transfers
GET	/api/transfers	Show transfers
GET	/api/transfers/id	show transfers by id (CURP)
DELETE	/api/transfers/id	to delete transfers by id (CURP)
PATCH	/api/transfers/	to update transfers

## Structure

📁 HH\_Backend

📁 Project

📁 Auth //to check authorization of users  
index.js

📁 Config //configuration of database  
dbConfig.js

📁 Controllers //to get the requested data from the models  
loginController.js  
patientController.js  
transferControllers.js  
usersController.js

📁 enums //to ensure that end-user has predefine options for certain fields  
index.js //User-roles Administrator and Responsible

📁 Routes //to forward the supported requests to the appropriated controller  
loginRoutes.js  
patientRoutes.js  
transferRoutes.js  
usersRoutes.js

📁 Services //contains the database queries and returning objects or throwing errors  
patientServices.js  
transferServices.js  
usersServices.js

📁 scratch //stores user session, it gets deleted once the user logs out

📁 node\_modules

.env


index.js


{ } package-lock.json

```
{ } package.json
```

## Schema

V 1.0.0

users
 id (a_u)
name
hospital
user
password
shift
createdAt
updatedAt

patients
 id
patient_name
patient_lastname
patient_second_lastname
patient_CURP (unique key)
patient_age
patient_gender
patient_birth
patient_address
patient_floor
patient_bed
patient_diagnostic
patient_status
patient_admission_date
hospital_id

transfers
 transfer_id
priority
date_sent
name_resp
lastN_resp
service_sent
diagnostic
clinic_sum
hospita_envia
hospita_recibe
patient_id

