# **Chapter 5 IMPLEMENTATION**

## 5.1 Source code

const {Sequelize,Op, QueryTypes} = require('sequelize');

const { Validator } = require('node-input-validator')

const db = require('../../models');

const argon2 = require('argon2')

const { v4: uuidv4 } = require('uuid');

const jwt = require('jsonwebtoken');

// multipleStatements: true -> set this in config file if you want to run mutiple queries

const {users, products, categories} = require('./mockups')

const {UsersInformation,Conversation, InventoryCategory,WishList,Inventory,TransactionReceipt,Transaction,DeliveryLocation} = require('../../models/index');

let generateToken = (email,userId) =>{

return jwt.sign({

emailAddress:email,

userId:userId

},

process.env.MY\_HASH\_SECRET);

}

var cloudinary = require('cloudinary').v2;

cloudinary.config = ({

cloud\_name:process.env.CLOUDINARY\_CLOUD\_NAME,

api\_key:process.env.CLOUDINARY\_API\_KEY,

api\_secret:process.env.CLOUDINARY\_API\_SECRET

})

exports.add\_new\_user = async (req,res,next)=>{

try {

const v = new Validator(req.body, {

firstName: "required|string|minLength:1",

lastName: "required|string|minLength:1",

email: "required|email",

password: "required|string",

phone: "required|phoneNumber",

isAdmin: "required|boolean",

isConsultant: "required|boolean"

})

const matched = await v.check()

if(!matched){

return res.status(412).json({

message:'Invalid Data Input'

});

}else{

let hashVerificationCode = await argon2.hash(req.body.password,process.env.MY\_ARGON\_SALT)

let userId = uuidv4();

let productUrl = null

if(req.files?.userImage){

let fileUpload = await cloudinary.uploader.upload(

req.files.userImage.tempFilePath,

{

folder: "pharmacy-products/"

})

productUrl = fileUpload.secure\_url

}

//create new user

let userInformation = await UsersInformation.create(

{

firstName : req.body.firstName.trim(),

profileImage : productUrl,

lastName : req.body.lastName.trim(),

email : req.body.email.trim(),

phone : req.body.phone.trim(),

userId:userId,

password : hashVerificationCode,

isConsultant : req.body.isConsultant,

isAdmin : req.body.isAdmin

});

const tokenValue = generateToken(userInformation.email,userInformation.userId)

return res.status(200).json({

message:'Created',

userInformation,tokenValue

});

}

} catch (error) {

return res.status(500).json({

message:'Fail',

error:error

});

}

}

exports.add\_new\_customer = async (req,res,next)=>{

try {

const v = new Validator(req.body, {

firstName: "required|string|minLength:1",

lastName: "required|string|minLength:1",

email: "required|email",

password: "required|string",

phone: "required|phoneNumber"

})

const matched = await v.check()

// console.log(v.errors)

// console.log(req.body)

if(matched){

let hashVerificationCode = await argon2.hash(req.body.password,process.env.MY\_ARGON\_SALT)

let userId = uuidv4();

//create new user

let userInformation = await UsersInformation.create(

{

firstName : req.body.firstName.trim(),

lastName : req.body.lastName.trim(),

email : req.body.email.trim(),

phone : req.body.phone.trim(),

userId:userId,

password : hashVerificationCode

});

const tokenValue = generateToken(userInformation.email,userInformation.userId)

return res.status(200).json({

message:'Created',

userInformation, tokenValue

});

}

return res.status(422).json({

message:'Fail'

});

} catch (error) {

return res.status(500).json({

message:'Fail',

error:error

});

}

}

**More on my Github Repo**

RestFul API:

React APP: