**Jobportal Project creation in Rails**

**Create a folder jobportal**

**d:\> md jobportal**

**d:\>cd jobportal**

[**d:\jobportal**](file:///d:/jobportal)>md jobbck

[**d:\jobportal**](file:///d:/jobportal)>cd jobbck

**To open Vscode**

**d:\jobportal\jobbck>code .**

**Create a new Project**

**rails new jobbck --database=postgresql**

**Create Landingpage routes:**

1. Go to config/routes.rb

root "application#landingpage"

2. Go to app/controllers/application\_controller.rb

Add a function called as hello inside the application controller class

def landingpage

render html: "welcome to our Jobportal"

end

**Execute rails app:**

**rails s -p 3001**

**Note: If no port is specified then it will run at 3000 port which is default.**

Since our react project is running in our port 3000. So we are using 3001 port for rails.

**Gem file:**

**Adding some required library files in gem file:**

**Open your gem file and uncomment the line no 37**

**gem "bcrypt", "~> 3.1.7"**

**then in terminal**

**D:\jobportal\jobbck>bundle install**

**After installing gemfile.**

**Now we are going to mention the database details in the database.yml file**

**Database.yml**

Go to config/database.yml

In Line no 26, mention your database name

database: jobportal3

uncomment line no 32

Mention your user name

username: postgres

uncomment line no 35

Mention your password

**password: password**

uncomment line no 40

Mention your host name . ( Here I have used amazonaws )

host: rubydb.c733iovakuat.us-east-1.rds.amazonaws.com

uncomment line no 44

port: 5432

**Create new controller:**

**rails generate controller useraccount**

**Go to app\controller**

**Now you are able to see , new controller useraccount\_controller.rb file**

**Create new migration file:**

**rails generate migration create\_useraccount**

Go to db\migration

Now you are able to see one migration file.

**Create new model file**

**rails generate model useraccount**

Go to app\models

now you are able to see new model file, useraccount.rb

Then open your useraccount migration file and give the table details

class CreateUseraccount < ActiveRecord::Migration[7.0]

def change

create\_table :useraccounts do |t|

t.string :email

t.string :password\_digest

t.string :usertype

t.timestamps

end

end

end

After entering table details , then execute the following command in command prompt

rails db:migrate

After this table created in database . To check this open another command prompt then check the directory.

Ensure that your rails application is running in the web and as well as your database server is on.

After checking directory , excute the following command

d:\jobportal\jobbck>rails c

Loading development environment (Rails 7.0.4.2)

irb(main):001:0>Useraccount.all

Useraccount Load (304.4ms) SELECT "useraccounts".\* FROM "useraccounts"

After checking this , write a API creation in controller

require 'bcrypt'

class UseraccountController < ApplicationController

skip\_before\_action :verify\_authenticity\_token

def index

render json: Useraccount.all

# current\_user =Useraccount.find\_by\_id(session[:current\_user\_id])

# render json: current\_user

end

def show

v = Useraccount.find(params[:id].to\_i)

render json: v

end

def create

v = !params[:email].empty? and !params[:password].nil?

if (v)

p=Useraccount.create(

'email': params[:email],

'password': params[:password],

'usertype': params[:usertype]

)

puts p

render json: "Data Added"

else

render json: "Data not added"

end

end

def update

p = Useraccount.find(params[:id].to\_i)

p.update(

'email': params[:email],

'password': params[:password],

'usertype': params[:usertype]

)

render json: "Data Updated"

end

def destroy

p = Useraccount.find(params[:id])

p.destroy

render json: "Data Deleted"

end

end

Then you can add records for this table.

Before adding records to the table

Open your post man and give your path

<http://127.0.0.1:3001/Useraccount>

use post method for update record

use get method for view record

use put method for edit record

use delete method for delete record

Do the same method for rest of the tables also

Table details

class CreateSkill < ActiveRecord::Migration[7.0]

def change

create\_table :skills do |t|

t.string :skillname

t.timestamps

end

end

end

class CreateJobdetail < ActiveRecord::Migration[7.0]

def change

create\_table :jobdetails do |t|

t.string :jobTitle

t.string :jobDescription

t.string :companyName

t.string :location

t.string :jobType

t.integer :salary

t.date :postedDate

t.string :domain

t.string :jobCode

t.string :skillsRequired

t.timestamps

end

end

end

class AddApplicationStatusToJobdetails < ActiveRecord::Migration[7.0]

def change

add\_column :jobdetails, :applicationStatus, :string

end

end

class CreateUserprofile < ActiveRecord::Migration[7.0]

def change

create\_table :userprofiles do |t|

t.string :firstName

t.string :lastName

t.string :email

t.string :contact

t.string :address

t.string :about

t.string :profilePic

t.string :currentCompany

t.integer :ctc

t.integer :experience

t.string :currentRole

t.string :skills

t.string :resumeLink

t.integer :expectedSalary

t.string :preferredLocation

t.timestamps

end

end

end

class CreateUserapplication < ActiveRecord::Migration[7.0]

def change

create\_table :userapplications do |t|

t.integer :userid

t.integer :jobid

t.string :jobCode

t.date :appliedDate

t.string :candidateApplicationStatus

t.string :location

t.timestamps

end

end

end

create a **login controller**

require 'bcrypt'

class LoginController < ApplicationController

skip\_before\_action :verify\_authenticity\_token

def create

usr = Useraccount.find\_by('email': params[:email])

if (usr.nil?)

render json: "User account does not exist"

else

if (usr.authenticate(params[:password]))

session[:current\_user\_id] = usr.id

render json: "Logged in successfully"

else

render json: "Incorrect Password"

end

end

end

def destroy

# session.delete(:current\_user\_id)

render json: "Logged out successfully"

end

end

Then create a **logout controller**

class LogoutController < ApplicationController

def index

session.delete(:current\_user\_id)

render json: "Logged out successfully"

end

end

Create a **method for authenticated user**

def authenticate\_user

authenticate\_user =Useraccount.find\_by\_id(session[:current\_user\_id])

end