UIT2602 WEB PROGRAMMING
Exercise 5 - To Integrate Devise for Authentication and Authorization

Name: S. Selcia
Reg. No.: 3122215002098
Section: IT B

1. **Aim:**

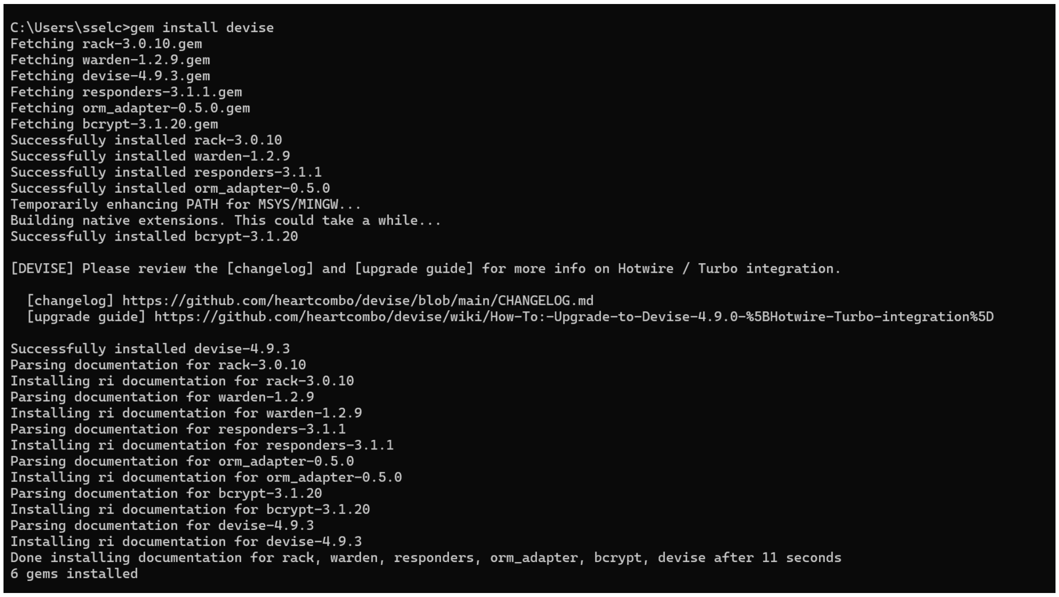
    Integrating Devise or another authentication gem into a Rails application is a common practice to handle user authentication and authorization.

2. **Required Web Tools and Methodology:**

    - Ruby on Rails: The web framework will be used to develop the application.
    - Web Browser: To test the functionality of the website.

3. **Output Screenshots:**

❖ The `*gem install devise*` command is used to add devise to our GemFile.



```
C:\Users\sselc>gem install devise
Fetching rack-3.0.10.gem
Fetching warden-1.2.9.gem
Fetching devise-4.9.3.gem
Fetching responders-3.1.1.gem
Fetching orm_adapter-0.5.0.gem
Fetching bcrypt-3.1.20.gem
Successfully installed rack-3.0.10
Successfully installed warden-1.2.9
Successfully installed responders-3.1.1
Successfully installed orm_adapter-0.5.0
Temporarily enhancing PATH for MSYS/MINGW...
Building native extensions. This could take a while...
Successfully installed bcrypt-3.1.20

[DEVISE] Please review the [changelog] and [upgrade guide] for more info on Hotwire / Turbo integration.

  [changelog] https://github.com/heartcombo/devise/blob/main/CHANGELOG.md
  [upgrade guide] https://github.com/heartcombo/devise/wiki/How-To:-Upgrade-to-Devise-4.9.0-%5BHotwire-Turbo-integration%5D

Successfully installed devise-4.9.3
Parsing documentation for rack-3.0.10
Installing ri documentation for rack-3.0.10
Parsing documentation for warden-1.2.9
Installing ri documentation for warden-1.2.9
Parsing documentation for responders-3.1.1
Installing ri documentation for responders-3.1.1
Parsing documentation for orm_adapter-0.5.0
Installing ri documentation for orm_adapter-0.5.0
Parsing documentation for bcrypt-3.1.20
Installing ri documentation for bcrypt-3.1.20
Parsing documentation for devise-4.9.3
Installing ri documentation for devise-4.9.3
Done installing documentation for rack, warden, responders, orm_adapter, bcrypt, devise after 11 seconds
6 gems installed
```

❖ Generate a controller file for the application using the command "*rails g controller home index*"
❖ Generate a model for the app using *rails generate model User name:string email:string password:string*
❖ Run the migration to create the database table rails db:migrate
❖ Run the migration to create the database table rails db:migrate
❖ Modify the config/routes.rb file to set the root route to the home index

```
Rails.application.routes.draw do get 'home/login'
get 'home/signup'
post 'home/signup', to: 'home#signup' post 'home/login', to: 'home#login' root 'home#index'
```

```
get 'home/secrets'
get "up" => "rails/health#show", as: :rails_health_check
end
```

- Modify the app/views/resumes/index.html.erb

```
<div class="container"> </br>
<h1>This is Home Page of this basic authentication app</h1> </div>
```

- Create and edit signin.html.erb

```
<div class="container">
</br>
<form action="/home/login" method="post"> <div class="form-group">
<label for="exampleInputEmail1">Email address</label>
<input type="email" class="form-control" id="exampleInputEmail1" name="email"
aria-describedby="emailHelp">
</div>
<div class="form-group">
<label for="exampleInputPassword1">Password</label>
<input type="password" class="form-control" id="exampleInputPassword1" name="password">
</div>
<button type="submit" class="btn btn-primary">Submit</button> </form>
</div>
```

- Create and edit signup.html.erb:

```
<div class="container"> </br>
<form action="/home/signup" method="post"> <div class="form-group">
<label for="exampleInputName1">UserName</label>
<input type="text" class="form-control" id="exampleInputName1" name="user[name]"
aria-describedby="emailHelp">
</div>
<div class="form-group">
<label for="exampleInputEmail1">Email address</label>
<input type="email" class="form-control" id="exampleInputEmail1" name="user[email]"
aria-describedby="emailHelp">
</div>
<div class="form-group">
<label for="exampleInputPassword1">Password</label>
<input type="password" class="form-control" id="exampleInputPassword1" name="user[password]">
</div>
<button type="submit" class="btn btn-primary">Submit</button> </form>
</div>
```

- Create users.db file:

```
class User < ApplicationRecord
has_secure_password
validates :name, presence: true
validates :email, presence: true, uniqueness: true, format: { with: URI::MailTo::EMAIL_REGEXP
}
validates :password, presence: true end
```

- Create _header.html.erb:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
<%= link_to 'Basic Authentication', root_path, class:"navbar-brand" %> <button
class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
<span class="navbar-toggler-icon"></span> </button>
<div class="collapse navbar-collapse" id="navbarNav"> <ul class="navbar-nav">
<li class="nav-item active">
<%= link_to 'Home', root_path, class:"nav-link" %> </li>

<li class="nav-item active">
<%= link_to 'Login', home_login_path, class:"nav-link" %> </li>
<li class="nav-item active">
<%= link_to 'Sign Up', home_signup_path, class:"nav-link" %> </li>
</ul> </div>
</nav>
```

- Create secrets.html.erb:

```
<div class="container">
</br>
<div class="jumbotron ">
<h1 class="display-4">Nice, You've executed it successfully</h1>
<p class="lead">This is a simple secret page for this basic authentication and authorization
app using ruby on rails.</p>
<hr class="my-4">
<p>By the way it took a lot of time to execute this functionality especially the bcrypt part
😭</p>
<%= button_to 'Try Again', root_path, method: :get, class: "btn btn-primary" %> </div>
</div>
```

- Create application.html.erb

```
<!doctype html> <html lang="en">
<head>
<!-- Required meta tags --> <meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<!-- Bootstrap CSS --> <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css"
integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
<title>Basic</title> <%= csrf_meta_tags %> <%= csp_meta_tag %>
<%= stylesheet_link_tag "application", "data-turbo-track": "reload" %> <%=
javascript_importmap_tags %>
</head> <body>
<%= render 'home/header' %>
```

```
<% flash.each do |key, message| %>
<div class="alert alert-<%= key %> alert-dismissible fade show" role="alert"> <%= message %>
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span> </button>
</div> <% end %>
<%= yield %>
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS --> <script
src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/js/bootstrap.min.js"
integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script>
</body> </html>
```

- Create home_controller.rb:

```
class HomeController < ApplicationController
before_action :require_login, only: [:secrets]
def index end
def signup
@user = User.new(user_params) if @user.save
Rails.logger.info "New user Created : #{@user}" redirect_to root_url, notice: "Sign up
successful!"
else
render :signup end
end
def login
email = params[:email] password = params[:password] loggedin = true
@user = User.find_by(email: email) puts "OK"
if @user && @user.authenticate(password) session[:user_id] = @user.id
puts "suc"
redirect_to '/home/secrets', notice: "Logged in successfully!" else
flash.now.alert = "Invalid email/password combination" unless loggedin puts "invalid
credentials"
render :login end
end

def secrets end
private
def user_params
params.fetch(:user, {}).permit(:name, :email, :password) end
def require_login
unless session[:user_id]
flash[:alert] = "You must be logged in to access this page." redirect_to "/home/login"
end end
end
```

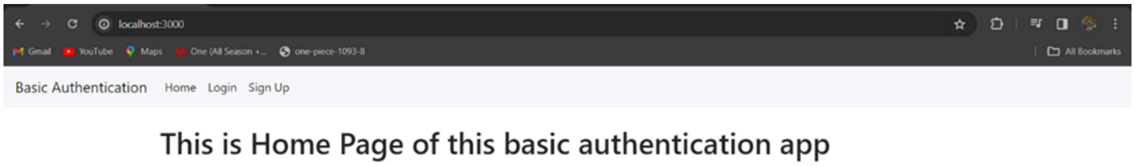
- Start the Rails server

```
rails s
```

- Now visit http://localhost:3000 in your browser to see your resume data displayed.

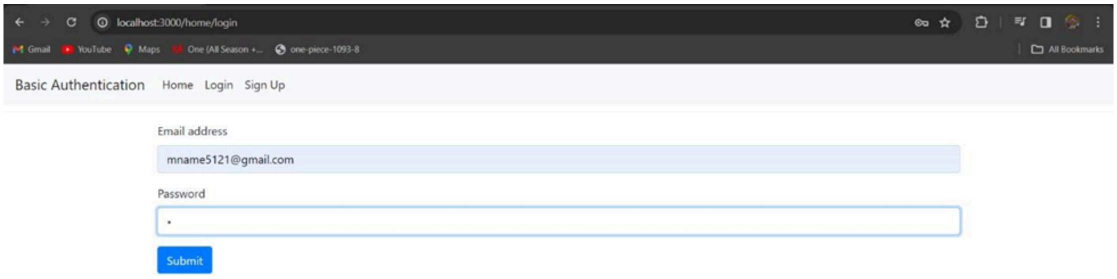
Sample Output:

Home page:



This is Home Page of this basic authentication app

- Login Page:



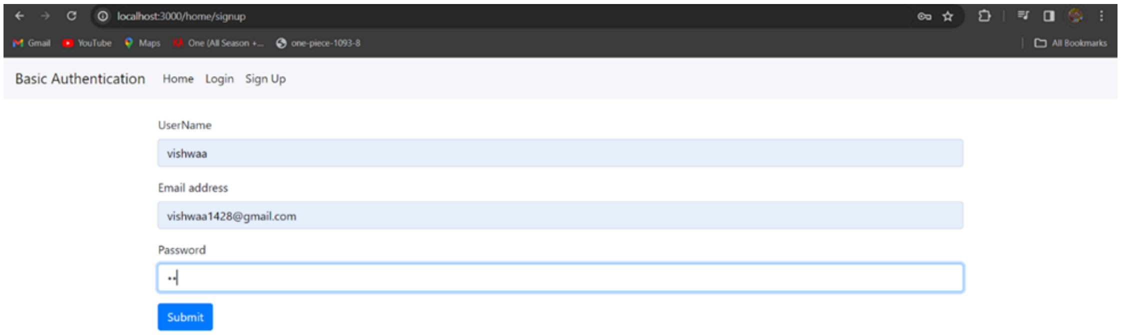
Email address

mname5121@gmail.com

Password

•

- SignUp Page:

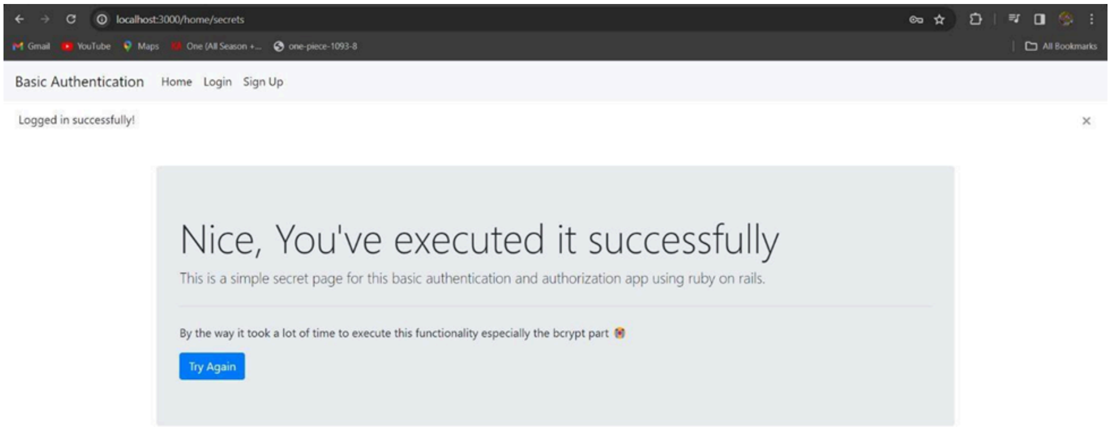


UserName
vishwaa

Email address
vishwaa1428@gmail.com

Password
••

- ❖ Secrets Page:



Logged in successfully!

Nice, You've executed it successfully

This is a simple secret page for this basic authentication and authorization app using ruby on rails.

By the way it took a lot of time to execute this functionality especially the bcrypt part

4. Conclusion:

In conclusion, integrating Devise or another authentication gem into a Rails application to handle user authentication and authorization has been implemented successfully.