\*\*모델생성

암호화된 패스워드 사용

회원관리

rails g model User name:string email:string password\_digest:string

\*\*db에서 중복안되는 값 받기

/Users/sangeun/lucy/users/db/migrate/20180308022415\_create\_users.rb

t.string :email, unique: true

rake db:migrate

\*\*유효성검사

/Users/sangeun/lucy/user/app/models/u.rb

class U < ApplicationRecord

before\_save { self.email = email.downcase }

VALID\_EMAIL\_REGEX = /\A[\w+\-.]+@[a-z\d\-]+(\.[a-z\d\-]+)\*\.[a-z]+\z/i

validates :name, presence: true, length: {maximum: 50}

validates :email, presence: true, length: {maximum: 255},

format: {with: VALID\_EMAIL\_REGEX}, uniqueness: {case\_sensitive: false}

end

presence 항상 존재 해야한다 / length최대 길이/ 유니크한가 /저장은 무조건 소문자로 저장된다

\*\*gemfile //gem ‘bcrypt’ 의 주석을 풀어주면 암호화된 비밀번호 사용가능

\*\* /Users/sangeun/lucy/user/app/models/u.rb

has\_secure\_password

추가 후 $ bundle

\*\* $ rails console –sandbox (콘솔 연습용)

User.create(name: "charles", email: "cyc@snu.ac.kr", password: "12345678", password\_confirmation: "12345678")

a=User.find(1)

\*\*\* irb(main):012:0> a.name

=> "charles"

irb(main):013:0> a.password

=> nil

irb(main):014:0> a.password\_digest

* "$2a$10$sUimLZk83lfJW5DOM0ZXsuIaegGoMdr7ukwp1PW26xlFGoTNjetx2"
* **authenticate 메소드**
* >> user.authenticate("1234") false >> user.authenticate("1234asdf") false
* 일반적으로 authenticate 에 실패했을 경우 false 를 반환합니다만,
* >> user.authenticate("12345678") => #<User id: 1, name: "charles", email: "cyc@snu.ac.kr", created\_at: "2016-05-01 11:26:43", updated\_at: "2016-05-01 11:26:43", password\_digest: "$3f$10$ss8d7f6f8VglgirzjSJOln.Fv9NdLMbqmR4rdTWIXY1G...">
* 성공했을 경우에는 위와 같이 자기 자신을 반환합니다. 하지만 authenticate 메소드에서 중요한 건 User 객체를 반환한 것이 중요한 것이 아니라 돌아온 값이 true 인 것이 중요합니다. true 값을 얻기 위해서는 authenticate 를 다음과 같이 사용하면 됩니다.
* >> !!user.authenticate ( "12345678") => true

!!user.authenticate ( "12345678") => true

\*\*컨트롤러 생성

rails g controller Users new

user controller에 new 자동완성 & view에 new html자동완성&route 파일에서도 get에 users파일이 생김

restful api를 만들때 생기는 경로를 모두 생성해준 것

/Users/sangeun/lucy/users/config/routes.rb

resources :users

user에 대한 경로 자동 생성 $ rake routes

<http://localhost:3000/users/new> 확인

/Users/sangeun/lucy/users/app/views/layouts/application.html.erb

body안에!

<a href="/users/new">회원가입</a>

<%= link\_to '회원가입', new\_user\_path %>

\*\*1. user모델 만들기

**직접 로그인 기능 구현해보기**

대부분의 웹 서비스는 사용자(유저)의 로그인 및 인증 시스템이 필요합니다. 이를 지원해주는 라이브러리는 Clearance, Authlogic, Devise, CanCan 등 여러 가지가 있습니다. 하지만 한번쯤은 직접 만들어보는 것이 좋습니다. 자신이 만든 시스템이라면 자신이 가장 잘 이해할 수 있고, 레일즈는 이 사용자 인증 시스템을 쉽게 만들 수 있도록 되어 있습니다. 다른 사람이 만든 라이브러리를 가져다 사용한 경우에도, 직접 이 사용자 인증 시스템을 구축해본 경험이 있다면, 보다 쉽게 이해하고 수정할 수 있게 될 것입니다.

**User 모델 설계**

User의 이름과 아이디로 사용될 이메일, 암호화된 패스워드를 저장할 공간을 만듭니다.

$ rails generate model User name:string email:string password\_digest:string

그리고 마이그레이션 파일을 토대로 데이터베이스를 생성해줍니다.

$ rake db:migrate

연습을 위해 콘솔을 실행해보겠습니다.

$ rails console --sandbox

--sandbox 옵션을 주게 되면 데이터베이스를 수정하더라도 실제 데이터베이스에 기록되지 않습니다. 연습용으로 좋습니다

**안전한 암호 적용하기**

다음은 최종 단계라고 말해도 좋은 "안전한 비밀번호"에 대해서 알아보도록 합니다. 비밀번호는 노출되어서는 안되고, 노출된다고 해도 암호화 되어 쉽게 알아볼 수 없도록 해야합니다. 이를 위해서 데이터베이스에 암호를 저장할 때는 **암호화 된 비밀번호**를 저장하게 되어 있습니다.

일반적으로 로그인 창에서 '12341234' 라는 비밀번호를 보내면 서버에서는 '12341234' 라는 암호와 비교하는 것으로 알고 있지만, 실제로 그렇지 않습니다. 송신된 암호를 암호화시키고, 데이터베이스의 암호화된 비밀번호와 비교한 후 인증하게 되어 있습니다. 이렇게하면 데이터베이스에 암호화되지 않은 평문의 비밀번호를 저장하는 위험한 일을 하지 않아도 됩니다.

**해시(암호화)된 암호**

말은 거창하게 했지만, 사실은 has\_secure\_password 라는 메소드를 User 모델에서 호출하면 거의 끝입니다. 다만 이 has\_secure\_password 를 사용하기 위해서는 몇 가지 조건을 갖추고 있어야 합니다.

* 데이터베이스의 password\_digest 라는 속성이 있어야하며 해당 속성에는 보안 해시된 암호가 저장된다.
* 두 쌍의 가상적인 속성을 사용할 수 있다. password, password\_confirmation. 두 속성은 회원가입 시 비밀번호와 비밀번호 재입력 값이 일치하는지 체크한다.
* authenticate 메소드를 사용할 수 있다.(사용자 인증하는 메소드입니다.)

**app/models/user.rb** 파일 수정

class User < ActiveRecord::Base has\_secure\_password end

**Gemfile 에서 bcrypt 주석 해제**

bcrypt 는 암호화와 관련된 Gem 이며, bcrypt 를 설치하여 has\_secure\_password 를 사용할 수 있게 됩니다.

source 'https://rubygems.org' . . . gem 'bcrypt', '~> 3.1.7' . .

그 후 $ bundle install 을 실행합니다.

**레일즈 콘솔 확인**

**유저 생성**

$ rails console >> User.create(name: "charles", email: "cyc@snu.ac.kr", ?> password: "12345678", password\_confirmation: "12345678") => #<User id: 1, name: "charles", email: "cyc@snu.ac.kr", created\_at: "2016-05-01 11:26:43", updated\_at: "2016-05-01 11:26:43", password\_digest: "$3f$10$ss8d7f6f8VglgirzjSJOln.Fv9NdLMbqmR4rdTWIXY1G...">

**해시된 암호 확인**

>> user = User.find\_by(email: "cyc@snu.ac.kr") >> user.password\_digest => "$3f$10$ss8d7f6f8VglgirzjSJOln.F4G//FGhyQSWCpghqRWQWITUYlG3XVy"

**authenticate 메소드**

>> user.authenticate("1234") false >> user.authenticate("1234asdf") false

일반적으로 authenticate 에 실패했을 경우 false 를 반환합니다만,

>> user.authenticate("12345678") => #<User id: 1, name: "charles", email: "cyc@snu.ac.kr", created\_at: "2016-05-01 11:26:43", updated\_at: "2016-05-01 11:26:43", password\_digest: "$3f$10$ss8d7f6f8VglgirzjSJOln.Fv9NdLMbqmR4rdTWIXY1G...">

성공했을 경우에는 위와 같이 자기 자신을 반환합니다. 하지만 authenticate 메소드에서 중요한 건 User 객체를 반환한 것이 중요한 것이 아니라 돌아온 값이 true 인 것이 중요합니다. true 값을 얻기 위해서는 authenticate 를 다음과 같이 사용하면 됩니다.

>> !!user.authenticate ( "12345678") => true

# \*\*2. 회원 가입 기능 구현

config/routes.rb 파일 수정

resources :users

app/views/layouts/application.html.erb 에 회원가입 링크 추가

<a href="/users/new">회원 가입</a>

또는

<%= link\_to '회원 가입', new\_user\_path %>

추가하기

app/views/users/new.html.erb 파일 만들기

<%= form\_tag ''/users" do %> 이메일 <input type="email" name="email"><br> 패스워드 <input type="password" name="password"><br> 패스워드 확인 <input type="password" name="password\_confirmation"><br> <input type="submit" value="회원가입"> <% end %>

app/controllers/users\_controller.rb 파일 수정

def create User.create(email: params[:email], password: params[:password], password\_confirmation: params[:password\_confirmation]) redirect\_to :root end