02190367
高柳奏和

課題 4-1

以下のコードを書いたが検索がうまくいかなかった

```ruby
require 'active_record'

ActiveRecord::Base.establish_connection(
  adapter: "sqlite3",
  database: "sandbox.db"
)

class Lab < ActiveRecord::Base
end

class LabMember < ActiveRecord::Base
end

get '/' do
  @title = 'Hello, world!'
  @world = 'WORLD!'
  erb :index
end

get '/lab_search' do
  if !Lab.where(:lab_name => params[:lab_name]).empty?
    @lab = Lab.where(:lab_name => params[:lab_name])
    @lab_members = LabMember.where(:lab_id => @lab.id)
  else
    redirect '/'
  end
  erb :lab_search
end

get '/member_search' do
  if !Lab.where(:member_name => params[:member_name]).empty?
    @lab = Lab.where(:member_name => params[:member_name])
    @lab_members = LabMember.where(:lab_name => @lab.lab_name)
  else
    redirect '/'
  end
end
```


課題 4-2

BASIC 認証：ユーザー名とパスワードを Base64 という変換方式を用いたデータとして送信する認証方式。Base64 は復号可能なので、通信傍受に対して弱い。

Digest 認証：ユーザー ID とパスワードを送信する点は BASIC 認証と同じだが、ハッシュ関数（MD5）を用いてそれらをハッシュにして送信するので、BASIC 認証よりも強固である。
短所は、すべての環境に対応しているわけではない。そのため、ユーザの環境がある程度わかっているうえで使用する必要がある。

課題 4-3

Web システムを外部から利用するためのプログラムの呼び出し規約の種類の一つで、REST と呼ばれる設計原則に従って策定されたもの。URL ですべてのリソースを一いに識別し、セッション管理や状態管理を行わない。同じ URL に対する呼び出しには常に同じ結果が返されることが期待される。リソースの操作は HTTP メソッドによって指定され、結果は HTML,XML,JSON 等で返される。結果は HTTP ステータスコードで通知する。

応用４－１

ユーザーTadokoroでログインした結果以下のようになった

# ユーザ: Tadokoro

| 日時 | 投稿者名 | メッセージ |
| --- | --- | --- |

## 入力フォーム

メッセージ　　投稿者名

Test　Tadokoro　Tadokoro　投稿