Download Soal pada : <https://thinfi.com/95vn> (Pass : 123/\*-)

1. Buat database dan import file .sql yang terdapat dalam folder database

\*Untuk kedepannya lakukan load helper, library, dan model menggunakan autoload.php

\*Semua assets yang digunakan sudah ada pada folder view, tugas kalian hanya memanggil halaman view tersebut, dan menulis syntax PHP CI nya

\***Autoload Model ‘User’ dan library ‘database’, set database.php pada folder config**

1. Ubah Controller default menjadi Controller **Landing**, ubah sehingga ‘Create an Account’ pada halaman View **Login** mengarah pada fungsi Register() pada Controller Landing, menggunakan Site\_url() **Gunakan ‘url’ helper**

A screenshot of a social media post

Description automatically generated



1. Pada halaman register, Buat sehingga ‘I am Already a Member’ mengarah ke fungsi Index() pada Controller Landing dan buat sehingga aksi yang dilakukan oleh form ketika di submit adalah memanggil fungsi Register() pada Controller UserController Menggunakan Site\_url(),
2. Ketika data disubmit, Pada fungsi UserController/Register(), lakukan pengecekan apakah kolom ‘password’ dan ‘re-enter password’ memiliki isi yang sama, jika **:**
   * **Tidak Sama (False),** Maka buatlah flashdata dengan nama ‘FailReg’ dan berisi ‘fail’, kemudian arahkan kembali ke Controller Landing Fungsi Register
   * **Sama (True),** Maka Panggil fungsi Register() pada Model User, untuk memasukkan data kedalam database, Jika fungsi Register Mengembalikan nilai :
     + **True,** Maka registrasi berhasil, buat flashdata ‘SuccessReg’ dan berisikan ‘Success’ dan arahkan kembali ke Controller Landing
     + **False,** Maka registrasi gagal, buat flashdata ‘FailReg’ dan berisikan ‘fail’ dan arahkan kembali ke Controller Landing fungsi Register

A screenshot of a cell phone

Description automatically generated

1. Pada Fungsi Register() di Model User, ambil inputan yang diberikan oleh user dari halaman Register, dan simpan kedalam variable array :

‘Username’ berisikan Username yang dimasukkan user,

‘Email’ Berisikan email yang dimasukkan user,

‘Pass’ Berisikan password yang dimasukkan user **(Password harus di enkripsi menggunakan MD5)*,***

***‘***RegisTime’ berisikan Tanggal dan waktu akun tersebut di daftarkan **(Waktu dan Tanggal harus sesuai dengan Waktu saat ini)**

Setelah itu panggil fungsi isExist() pada Model User yang **berfungsi melakukan pengecekan apakah Username yang diberikan sudah terdaftar** pada database, Jika :

* **Username sudah terdaftar,** Kembalikan True
* **Username belum terdaftar,** Kembalikan False

Jika hasil dari pemanggilan fungsi isExist() pada fungsi Register() di Model User adalah :

* **True,** Maka berarti data sudah ada, dan Register() mengembalikan nilai False
* **False,** Maka berarti data belum ada, Inputkan data kedalam table ‘Register’ dan table ‘Login’, kemudian Register() mengembalikan nilai True

1. Pada Halaman View ‘Login’ :

* **Jika Berhasil Registrasi :** Tampilkan Alert dari halaman View ‘Alerts/SuccessRegis’ pada bagan kode yang diberikan, dengan mengecek apakah Flashdata ‘SuccessReg’ ada atau tidak
* **Jika Gagal Registrasi :** Tampilkan Alert dari halaman View ‘Alerts/FailLogin’ pada bagan kode yang diberikan, dengan mengecek apakah Flashdata ‘FailReg’ ada atau tidak
* Buatlah sehingga form pada halaman login, ketika di submit, akan memanggil fungsi Signin() pada Controller UserController

1. Pada fungsi Signin(), Controller UserController :

* Panggil fungsi findUser() pada Model User dan simpan pada **$data**; findUser() berfungsi untuk menemukan data yang diinputkan oleh user (inputan username dan password oleh user dengan data pada database table ‘Login’) dan mengembalikan hasilnya
* Jika **$data** :
  + **Tidak memiliki isi** : Berarti data untuk login yang dimasukkan user tidak ditemukan, buatlah flashdata dengan nama ‘falselogin’ dengan isi ‘nodata’, dan arahkan kembali ke controller Landing
  + **Jika data ditemukan** : Berarti data untuk login ditemukan, lakukan pengecekan apakah user menceklist tombol remember-me ketika melakukan login, Jika :
    - **Remember-me di Checklist :** buatlah cookie bernamakan ‘logged’ dengan isi Username dari user yang melakukan Login, dengan jangka hidup cookie 3600, **Gunakan helper cookie**
    - **Remember-me tidak di Checklist :** Buatlah session bernamakan ‘successLogin’ dengan isi Username dari user yang melakukan login, **Gunakan Library Session**

1. Pada Controller Landing :

* Pada fungsi index() :
  + Ambil / fetch cookie ‘logged’
  + **Jika cookie ‘logged’ atau Session ‘successLogin’ ditemukan / diset, maka** panggil fungsi loginPage() pada controller Landing; fungsi loginPage() berisikan :
    - **Memanggil fungsi getImage() pada Model User dan simpan hasil pemanggilan kedalam variable $data :** getImage() akan mengembalikan semua data pada table ‘image\_path’, data adalah nama file dari foto dummy
    - Lalu memanggil halaman view ‘loggedin’ dan lakukan *passing* data $data ke view ‘loggedin’

1. Pada halaman view ‘loggedin’ :

* Buatlah sehingga tombol logout mengarah kepada fungsi logout() pada Controller UserController
* Pada line 16 – 19 :
  + Fetch Cookie yang dibuat
  + **Jika cookie ada : Maka echo isi dari cookie, atau**
  + **Jika session ada : Maka echo isi dari session**
* Pada line 55 :
  + Buat sehingga data yang di fetch dari fungsi getImage() bisa ditampilkan disini; (gunakan perulangan) direktori dari file dummy foto terdapat di ‘./uploads’,

1. Pada halaman view ‘loggedin’ :

* Buat sehingga form yang diberikan mengarah kepada fungsi do\_upload() pada Controller Uploads, menggunakan **form helper**
* Pada fungsi do\_upload() pada Controller Uploads :
  + Ubah sehingga file yang di upload akan disimpan pada folder ‘./uploads/’
  + Ubah sehingga format file yang diterima hanya ‘gif’ ‘jpg’ dan ‘png’
  + Ubah sehingga ukuran file yang bisa diupload hanya max. 4096 kB
  + Ubah sehingga lebar file yang bisa dikirim hanya 1920px
  + Ubah sehingga tinggi file yang bisa dikirim hanya 1080px
  + Lakukan proses upload, Jika :
    - **Upload Tidak Berhasil :** buat flashdata bernama ‘error’ berisikan umpan balik error dari upload tersebut
    - **Jika Upload Berhasil :**
      * Ambil informasi lengkap dari file yang diupload
      * Buatlah flashdata dengan nama ‘upload\_data’ berisikan nama file yang diupload
      * Panggil fungsi imageUpload() pada Controller User; imageUpload() menerima inputan nama dari file yang diupload disimpan kedalam variable $imgName, kemudian nama tersebut dimasukkan kedalam database table ‘image\_path’
      * Arahkan kembali ke Controller Landing/loginPage
* Buatlah alert yang menandakan berhasil upload atau tidak, berdasarkan session yang sudah dibuat pada fungsi do\_upload()

1. Lakukan percobaan upload pada halaman loggedin, jika program berjalan dengan baik maka gambar baru yang diupload akan tampil pada tampilan gallery dibawahnya.

A screenshot of a social media post

Description automatically generated

Tampilan Normal

A screenshot of a social media post

Description automatically generated

Tampilan Gagal Upload

A screenshot of a social media post

Description automatically generated

Tampilan Berhasil Upload