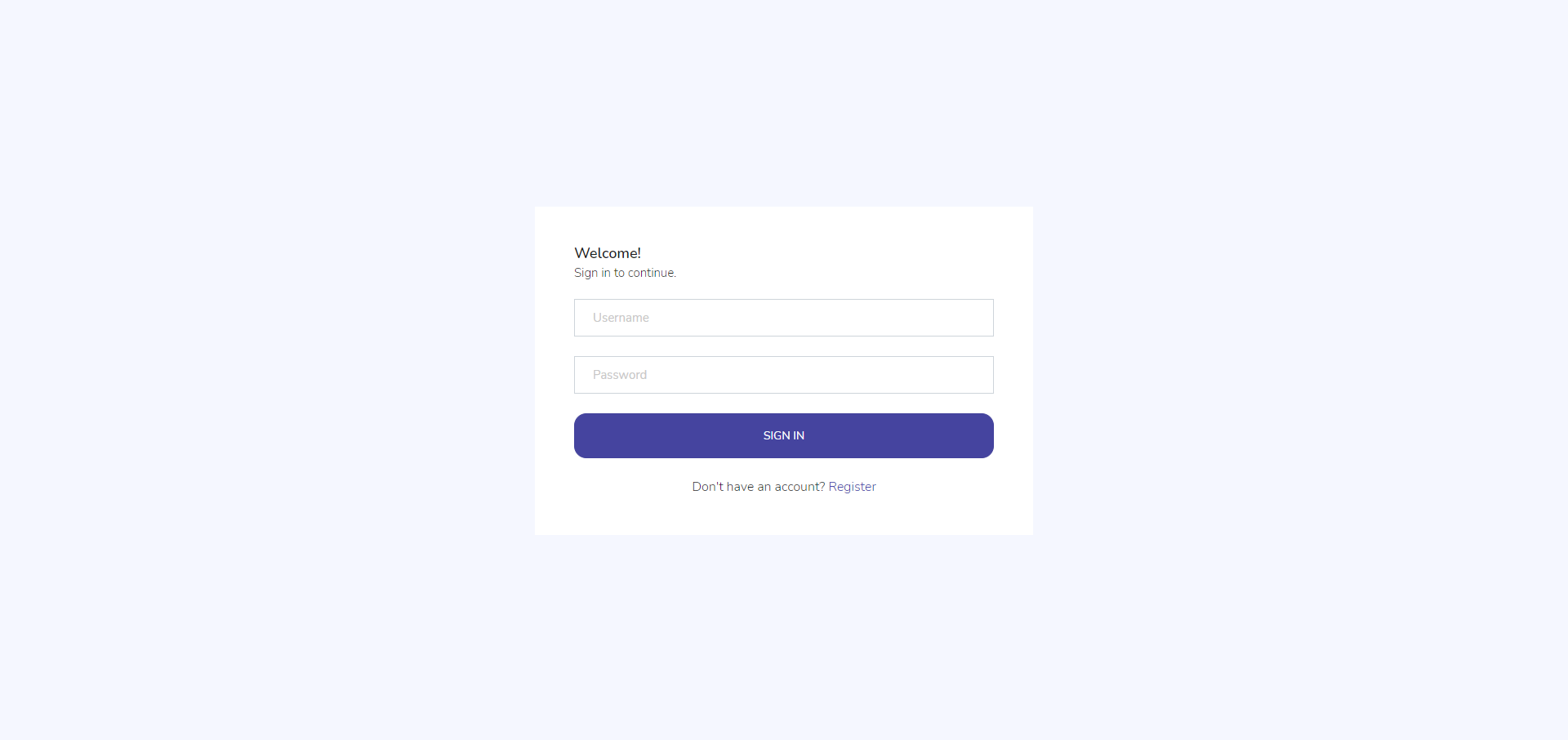
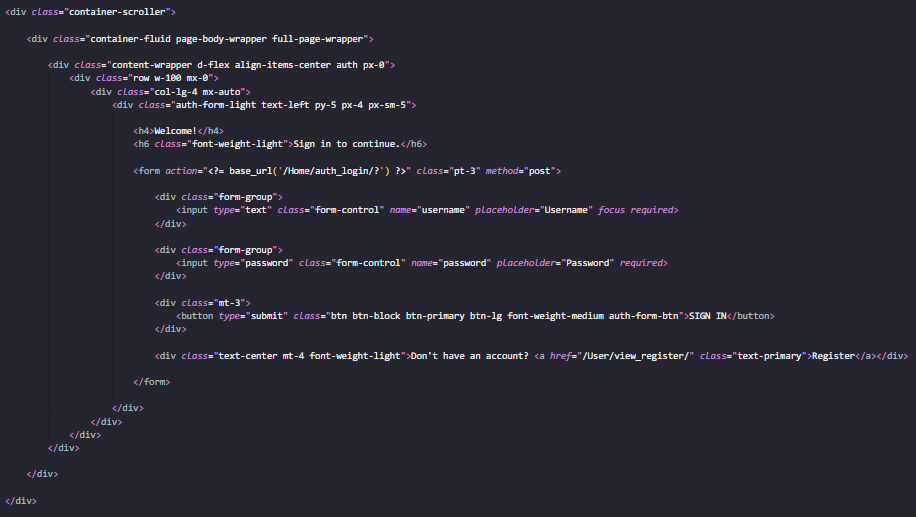
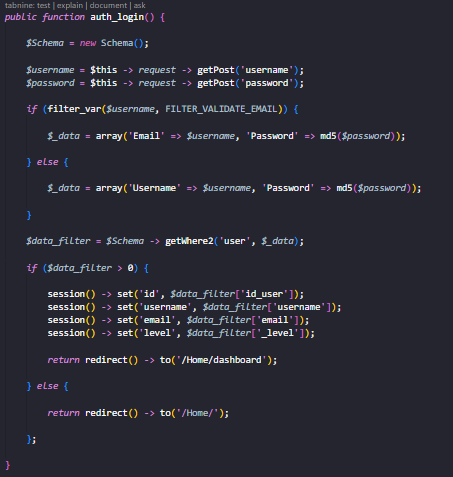
1. **Login**

****

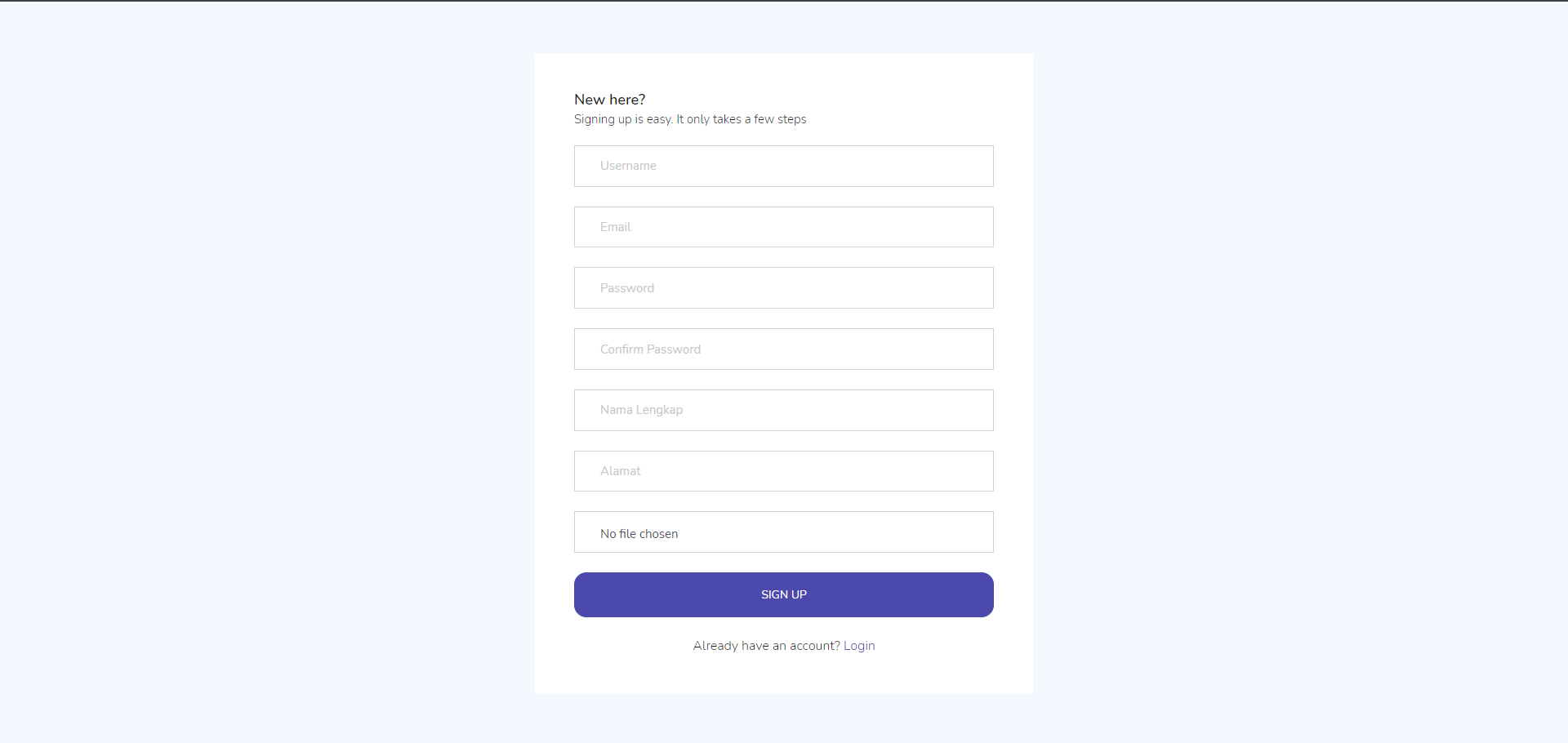
****

Pada halaman login terdapat input form untuk memasukkan **username** dan **password** untuk melakukan login terhadap website gallery foto, namun jika pengguna tidak memiliki akun, bisa terlebih dahulu melakukan registrasi dengan menekan tulisan **Register** yang berada di bawah tombol **SIGN IN** yang berfungsi untuk mengirim data kepada form yang dituju.

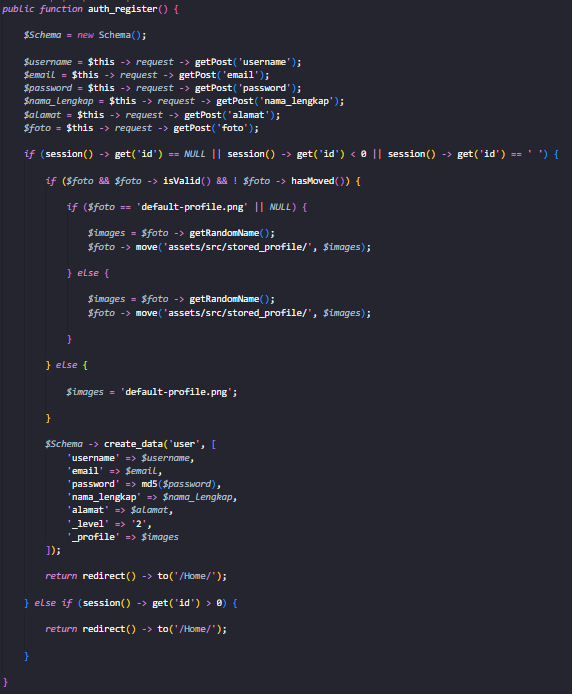


Pada controllernya terdapat fungsi **auth\_login()** yang dimana fungsi ini digunakan untuk melakukan proses autentikasi pengguna yang ingin masuk ke dalam website ini, fungsi ini akan mengambil data **username** dan **password** yang dimana usernamenya juga menggunakan validasi terhadap input berupa **email**, data yang di ambil merupakan value dari tags **input** pada view login, selanjutnya data yang sudah di ambil lalu dikirim melalui form login. Lalu tujuan dari dibuatnya instance model **Schema()** yaitu sebagai sarana interaksi dengan database. Selanjutnya data yang telah di ambil tadi kemudian di masukkan sebagai array yang lalu data tersebut dilakukan validasi terhadap table user menggunakan fungsi **getWhere2()** dari model **Schema()** dimana tujuannya untuk mencari data pengguna yang melakukan login tersebut. Jika datanya valid pada database, maka selanjutnya pengguna akan ditetapkan session yang dimana terdapat (id, username, email, level) lalu pengguna akan di arahkan ke halaman **dashboard**. Namun jika pengguna tidak terdeteksi maka pengguna akan secara otomatis di arahkan kembali ke halaman **login**.

1. **Register**

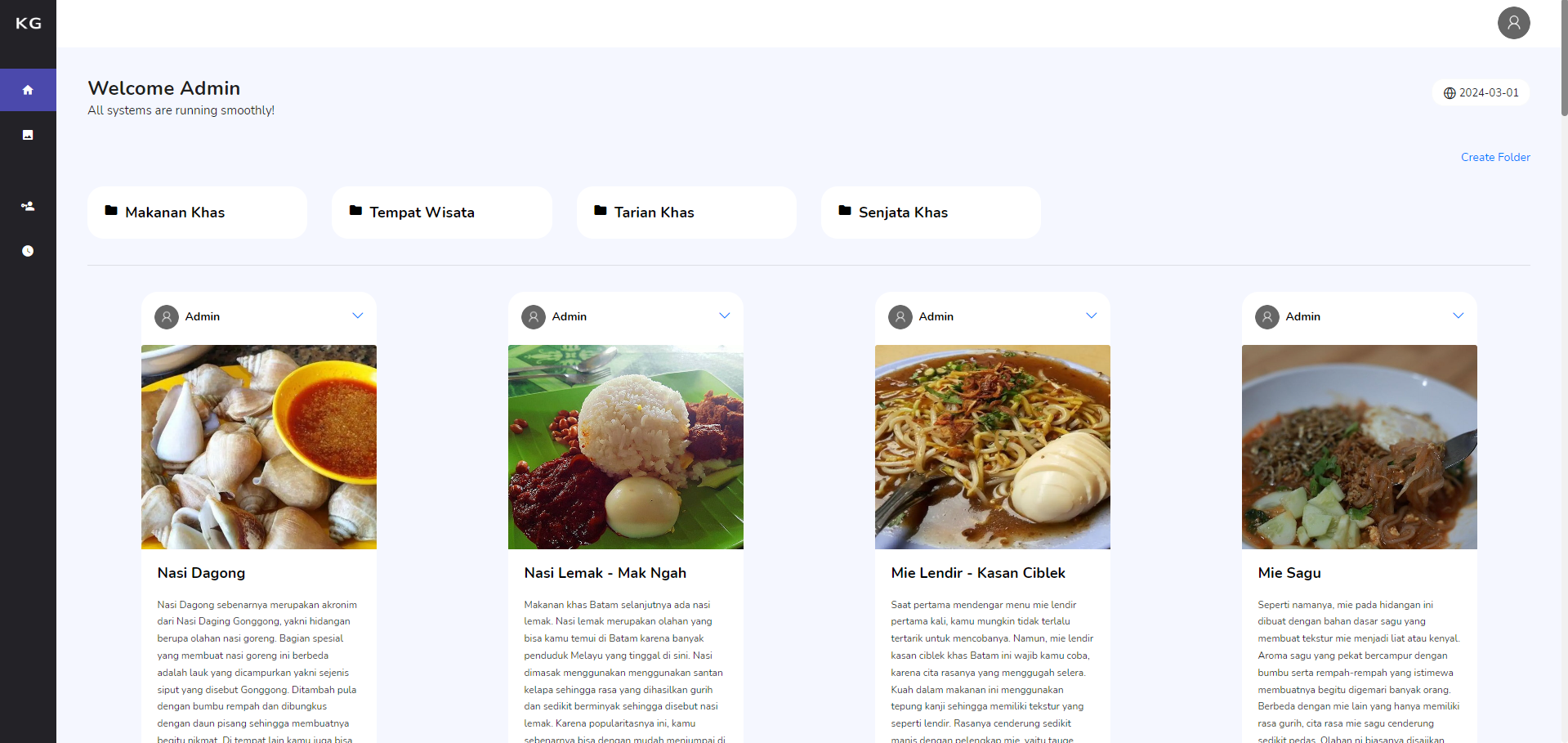
****

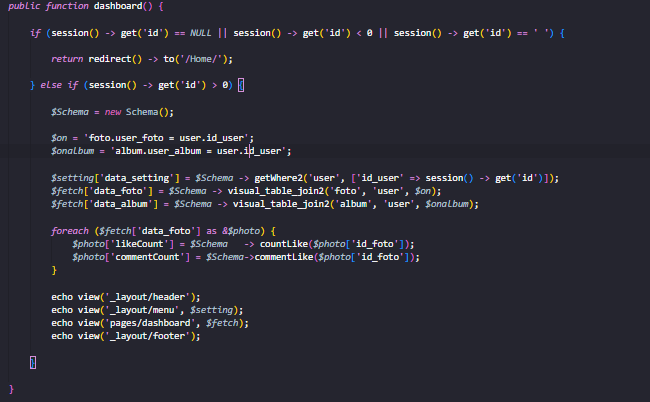
pada halaman register terdapat input form yang digunakan untuk memasukkan data berupa **username**, **email**, **password**, dan juga pengguna wajib melakukan konfirmasi password, **nama lengkap**, **alamat**, beserta **foto profil**, tujuan dari register ini yaitu untuk melakukan login agar mendapatkan akses ke dalam website gallery foto ini. Jika pengguna sudah melakukan input data terhadap form yang telah disediakan selanjutnya, user harus menekan tombol **SIGN UP** agar data yang telah di input tadi masuk ke dalam database. Setelah itu pengguna akan langsung di arahkan ke halaman **login** kembali untuk melakukan login.

****

Pada controllernya terdapat fungsi **auth\_register()** yang dimana fungsi ini digunakan untuk melakukan proses autentikasi pengguna yang ingin registrasi akun di website ini, fungsi ini akan mengambil data **username**, **email**, **password**, **nama lengkap**, **alamat**, dan juga **foto profil**. Data yang di ambil merupakan value dari tags **input** pada view register, selanjutnya data yang sudah di ambil lalu dikirim melalui form registrasi. Selanjutnya data yang telah di ambil tadi kemudian di masukkan sebagai array yang lalu data tersebut dilakukan input terhadap table user menggunakan fungsi **create\_data()** dari model **Schema()** dimana tujuannya untuk menambahkan data sesuai dengan data yang telah di masukkan. Disini juga terdapat validasi terhadap pengguna dimana jika pengguna tidak memiliki session id yang sedang berjalan maka akan dilanjutkan proses penginputan data ke dalam database, namun jika pengguna memiliki session id maka pengguna tersebut akan di arahkan kembali ke halaman dashboard melalui class **/Home/**.

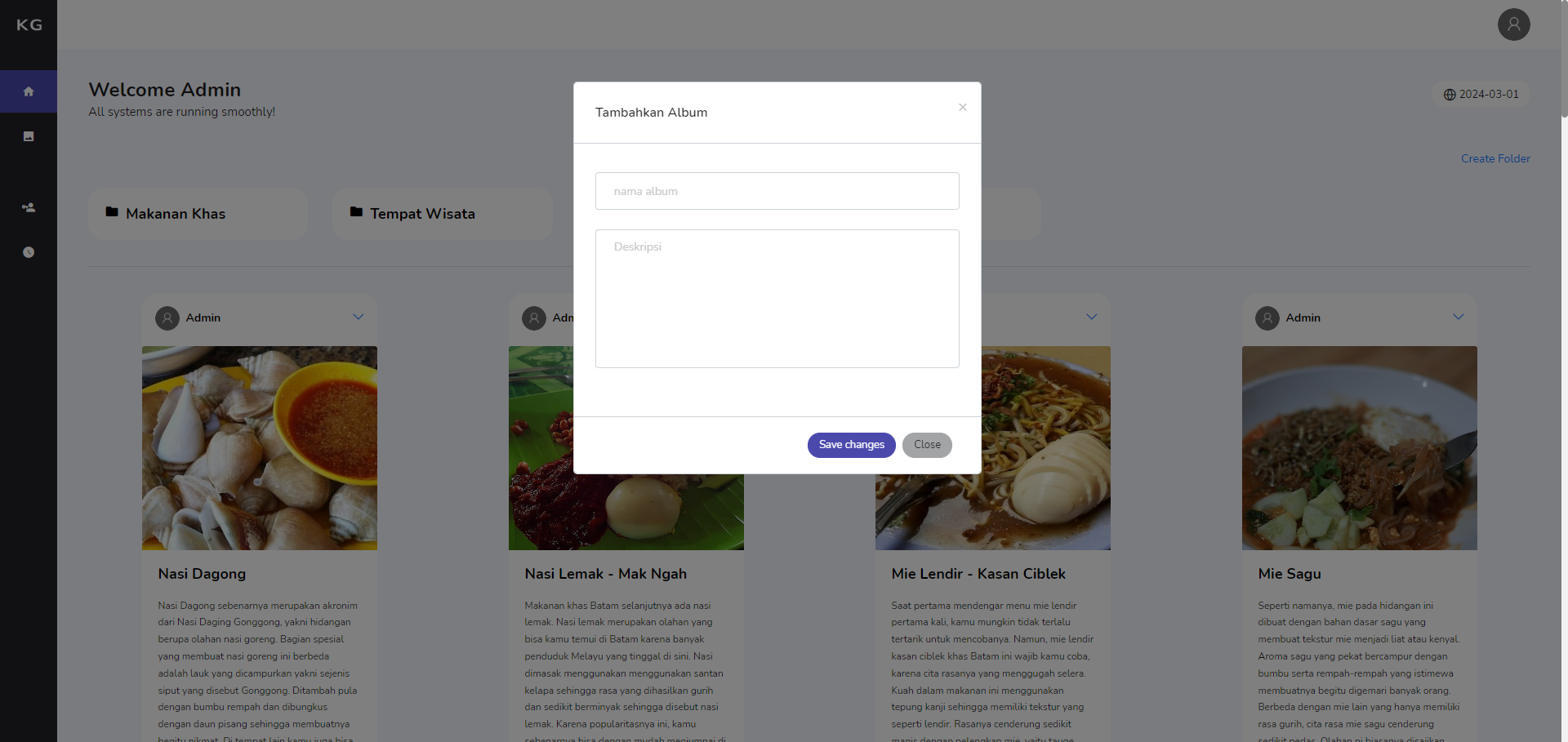
1. **Dashboard**

****

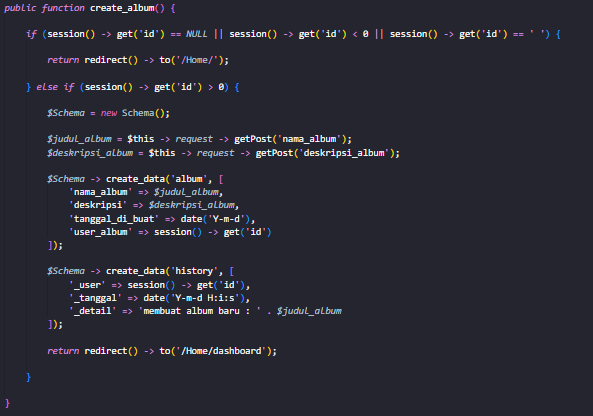
****

Pada controllernya terdapat fungsi **dashboard()** yang dimana fungsi ini digunakan untuk menampilkan halaman dashboard, dimana setiap pengguna yang telah berhasil melakukan login akan diarahkan ke halaman ini. Pada dashboard terdapat data yang ditujukan untuk menampilkan data berupa foto dan juga album beserta nama pemiliknya, yaitu dengan menggunakan fungsi **visual\_table\_join2()** yang dimana fungsi ini dipakai untuk melakukan join terhadap 2 table. Lalu terdapat juga method foreach yang digunakan untuk menghitung berapa banyak jumlah like beserta comment pada setiap image.

1. **Membuat Album**

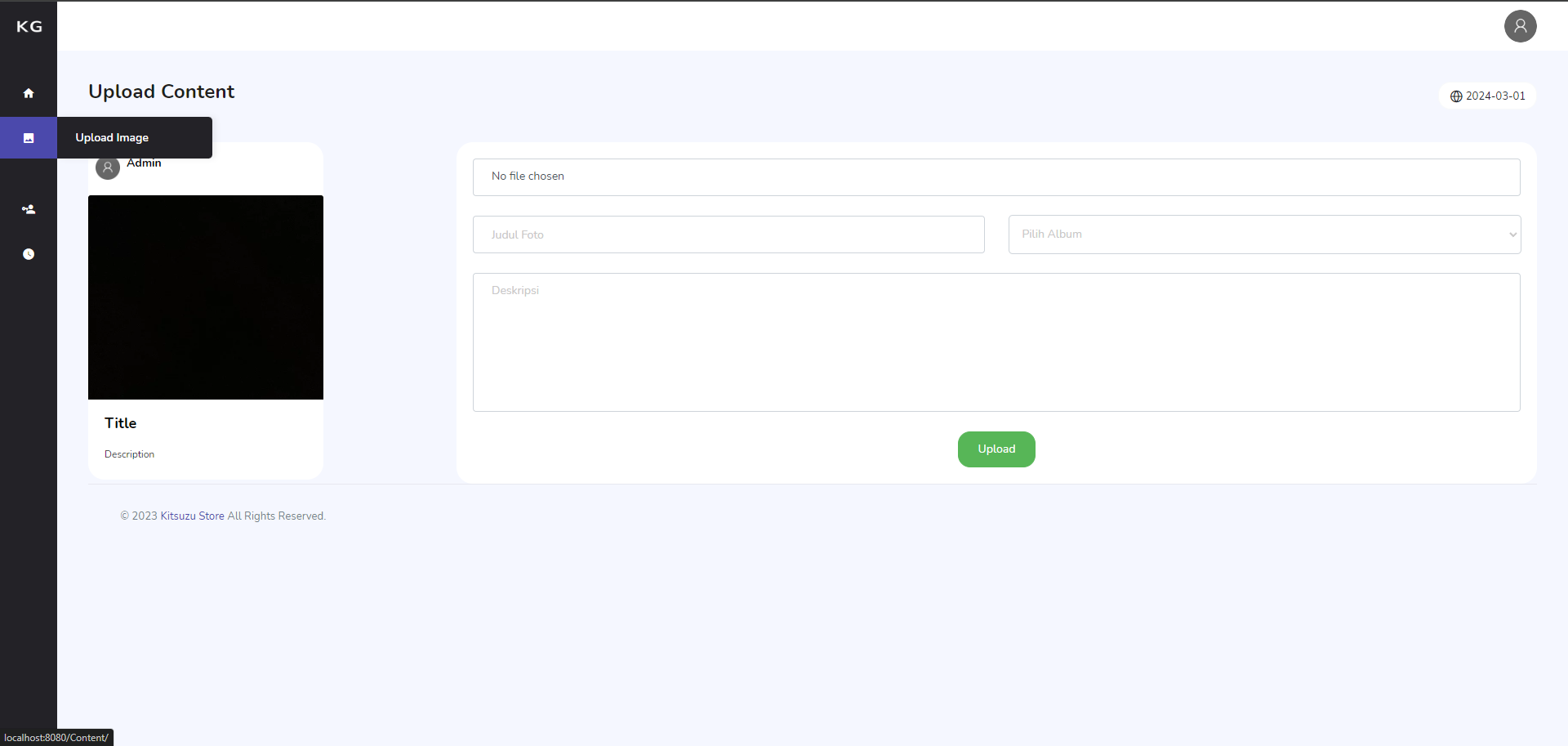
****

Pada halaman dashboard ini terdapat button pada kanan atas yang ditujukan untuk memanggil modals untuk menambahkan data album. Lalu pengguna bisa menekan tombol **Save Changes** ketika sudah selesai menginput data album yang akan di tambahkan.

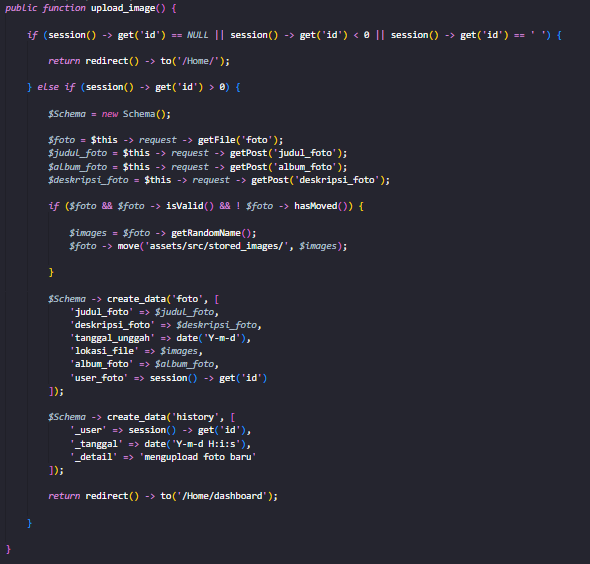
****

Pada controllernya terdapat fungsi **create\_album()** yang dimana fungsi ini digunakan untuk menginput data album yang akan dibuat. Pada fungsi ini terdapat data yang di ambil yaitu berupa **nama album** dan **deskripsi album** yang lalu data tersebut dimasukkan ke dalam database menggunakan fungsi **create\_data()** yang tablenya di arahkan ke table album yang dimana akan diisi dengan data yang sudah array. Lalu pada fungsi ini juta terdapat sistem penambahan history, yang dimana setiap kali pengguna membuat album baru maka akan di catat, dimana history ini mencakup **user**, **tanggal**, dan juga **detail** dari data yang dimasukkan.

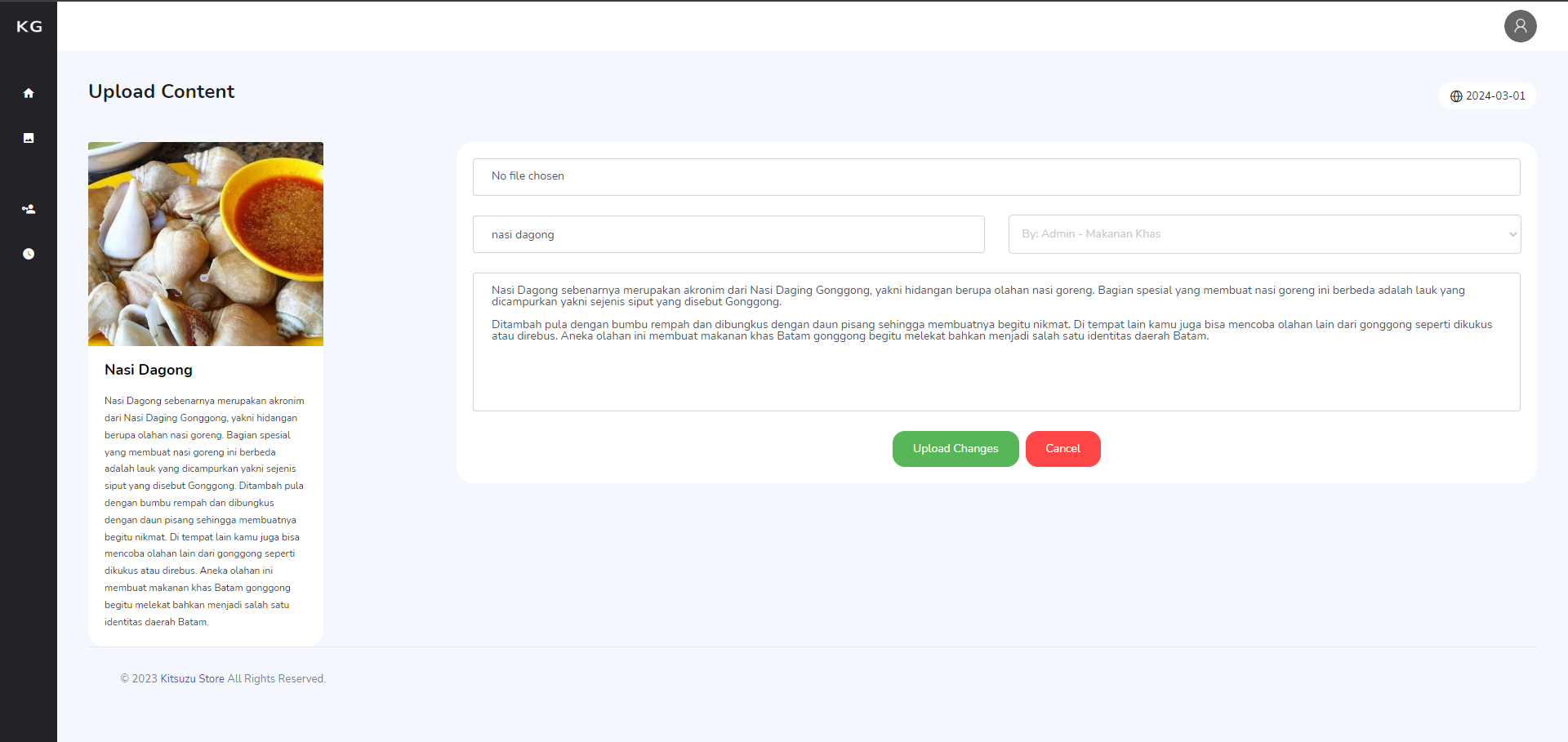
1. **Menambahkan Foto**

****

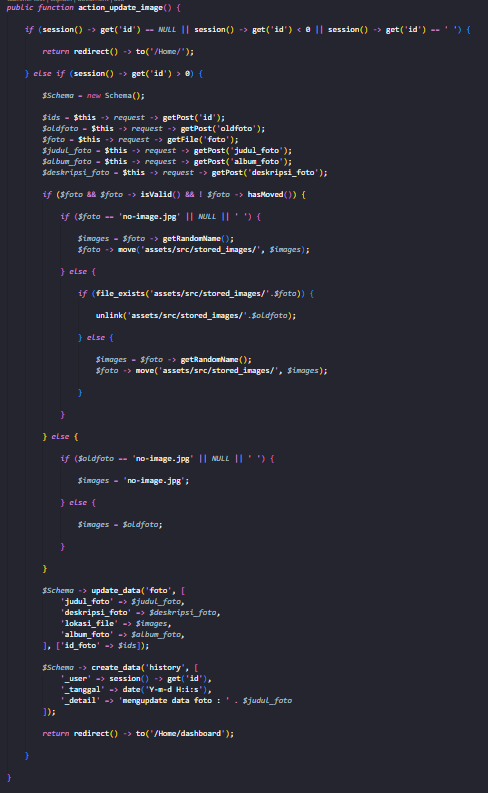
Pengguna dapat pergi ke halaman upload image untuk menambahkan postingan baru yang dimana pada halaman ini terdapat preview dari data yang akan tampil saat sudah selesai diinput, pada halaman ini pengguna dapat mengunggah foto yang akan di post, judul dari postingan, lalu album dari postingan, beserta deskripsi dari postingan. Lalu semua datanya akan di jadikan ke dalam array dan di upload dengan menggunakan function **upload\_image()**

****

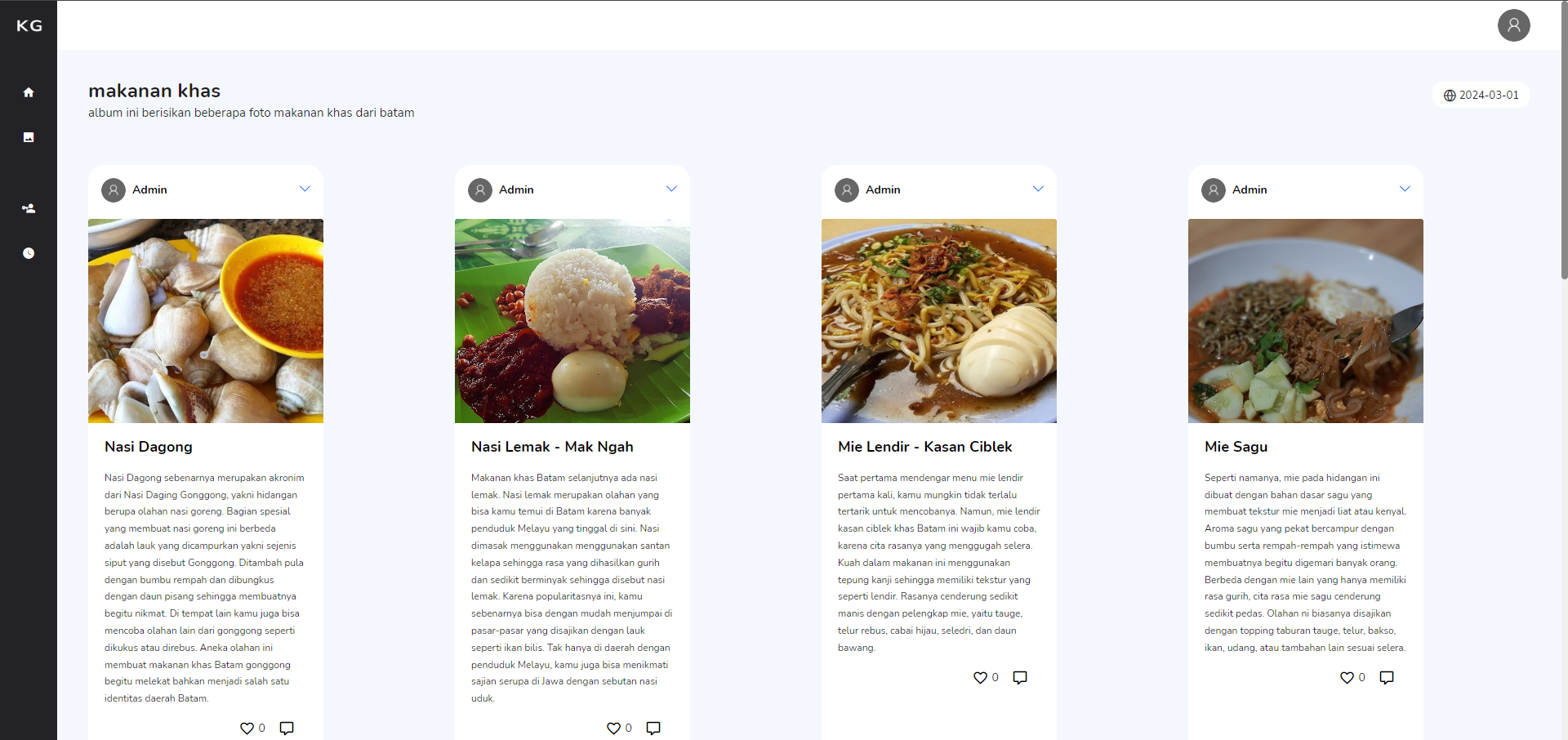
1. **Mengedit Foto**

****

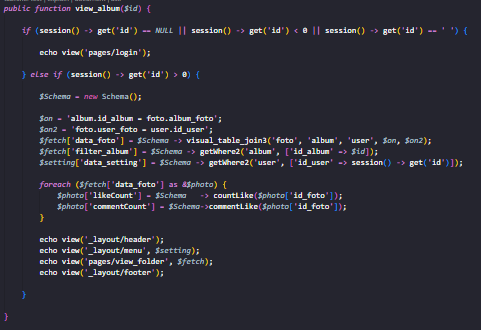
Pada halaman edit foto ini menggunakan pemanggilan data menggunakan getWhere yang dimana datanya diperoleh sesuai dengan id\_foto yang di pilih pada halaman dashboard / album. Di halaman ini pengguna dapat mengubah data yaitu seperti foto, judul postingan, album postingan, dan juga deskripsi postingan. Yang lalu datanya akan di jadikan array pada controller fungsi **update\_user()** dan untuk dapat mengupdate data foto ini diperlukan id\_fotonya lagi dengan menggunakan input type hidden yang valuenya merupakan id\_foto. Setelah itu semua datanya akan diperbaharui dengan menggunakan fungsi **update\_data()** dengan nama table sebagai **foto** dan data yang akan diubah sebagai array, beserta id\_foto yang akan digunakan untuk mencari data pada database yang akan diubah. Pada halaman ini juga terdapat validasi untuk foto yang akan di upload pada storage assets website ini, beserta mencatat history perubahan yang dilakukan untuk data **foto**.

****

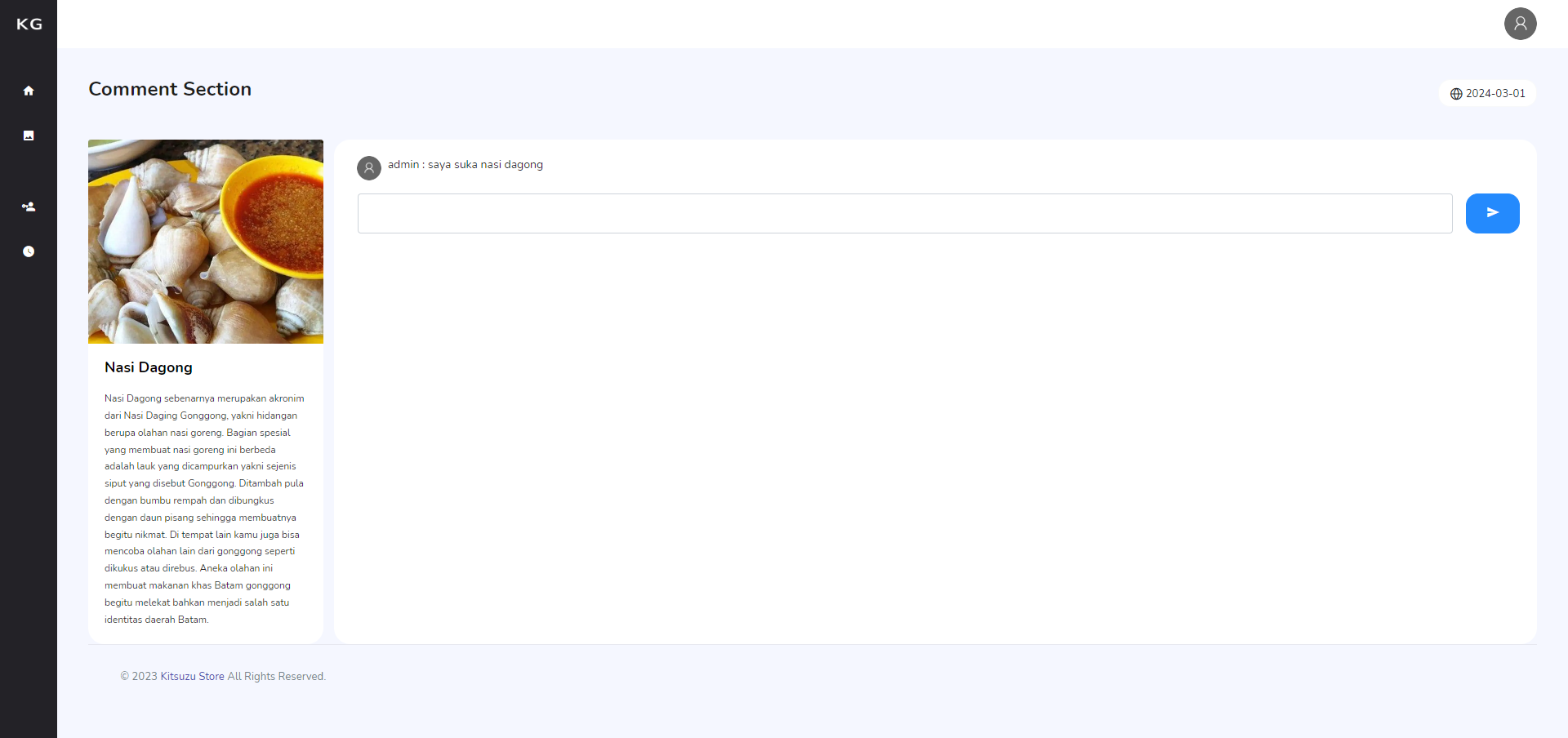
1. **Album**

****

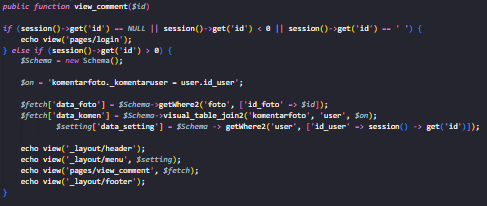
Pada halaman ini menampilkan seluruh postingan berdasarkan albumnya, dengan cara yaitu seperti function **view\_album()** dibawah yang dimana menggunakan fungsi **visual\_table\_join3()** dari instance **Schema()** yang berfungsi untuk melakukan join 3 table, pada website ini diambil data berdasarkan table foto, album, dan user, dan juga terdapat fungsi **getWhere2()** yang memakai table album sebagai filter di mana foto yang nantinya di ambil kan akan berdasarkan **foto.album\_foto** == **album.id\_album**

****

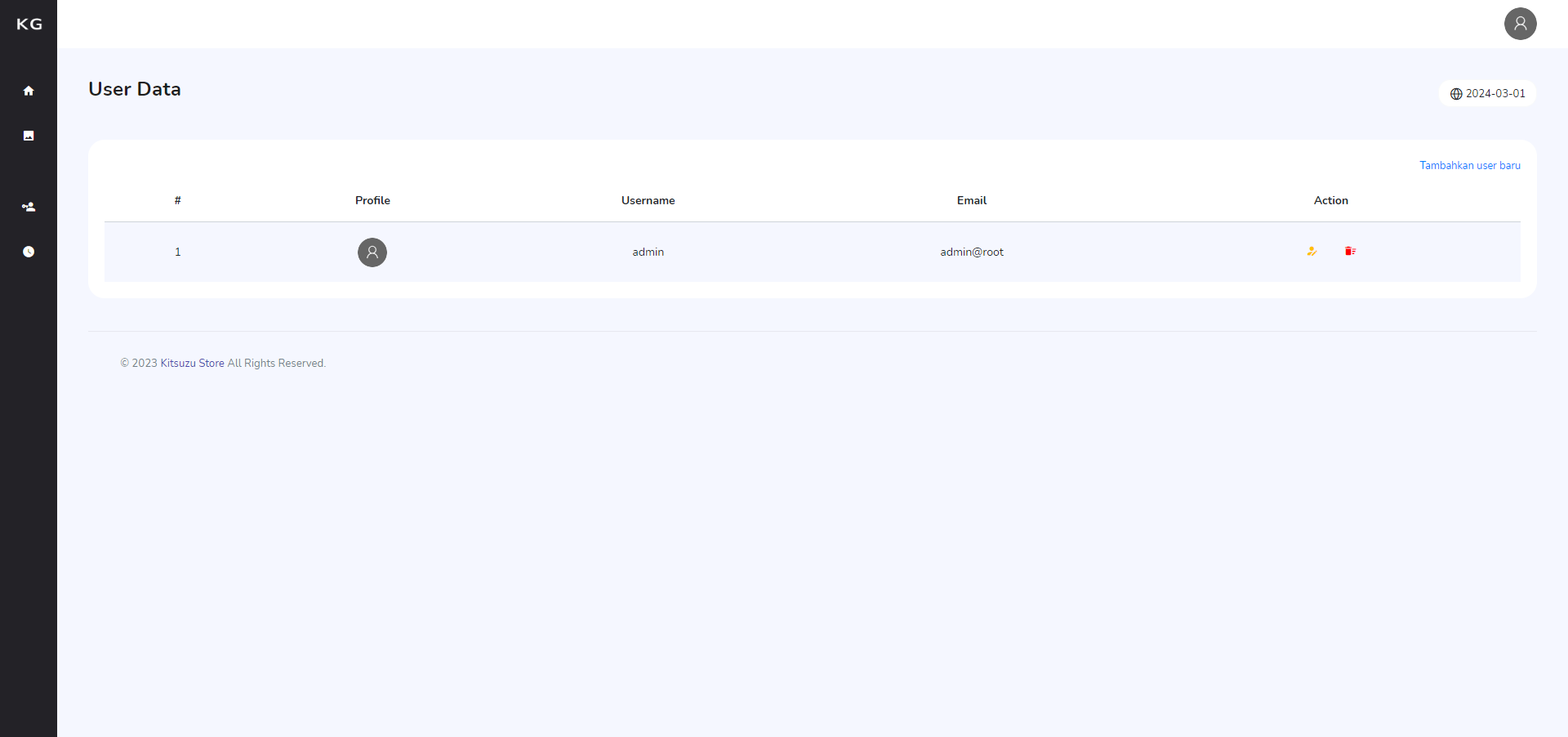
1. **Komentar**

****

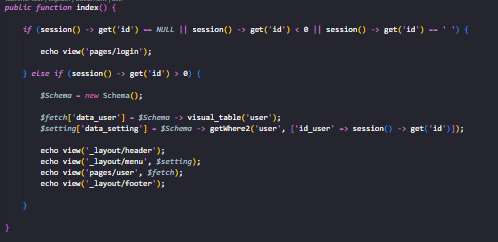
Pada halaman komentar ini pengguna dapat memberikan komentar mengenai postingan yang telah di posting oleh penggunanya sendiri maupun pengguna yang lain. Dimana semua komentar di peroleh dengan menggunakan pemanggilan data menggunakan fungsi **visual\_table\_join2()** dimana digunakan join berdasarkan table komentarfoto yang dijoin dengan table user melalui **komentarfoto.\_komentaruser** = **user.id\_user**. Dan pengguna dapat menambahkahkan komentar dengan input form yang telah disediakan pada halaman website.

****

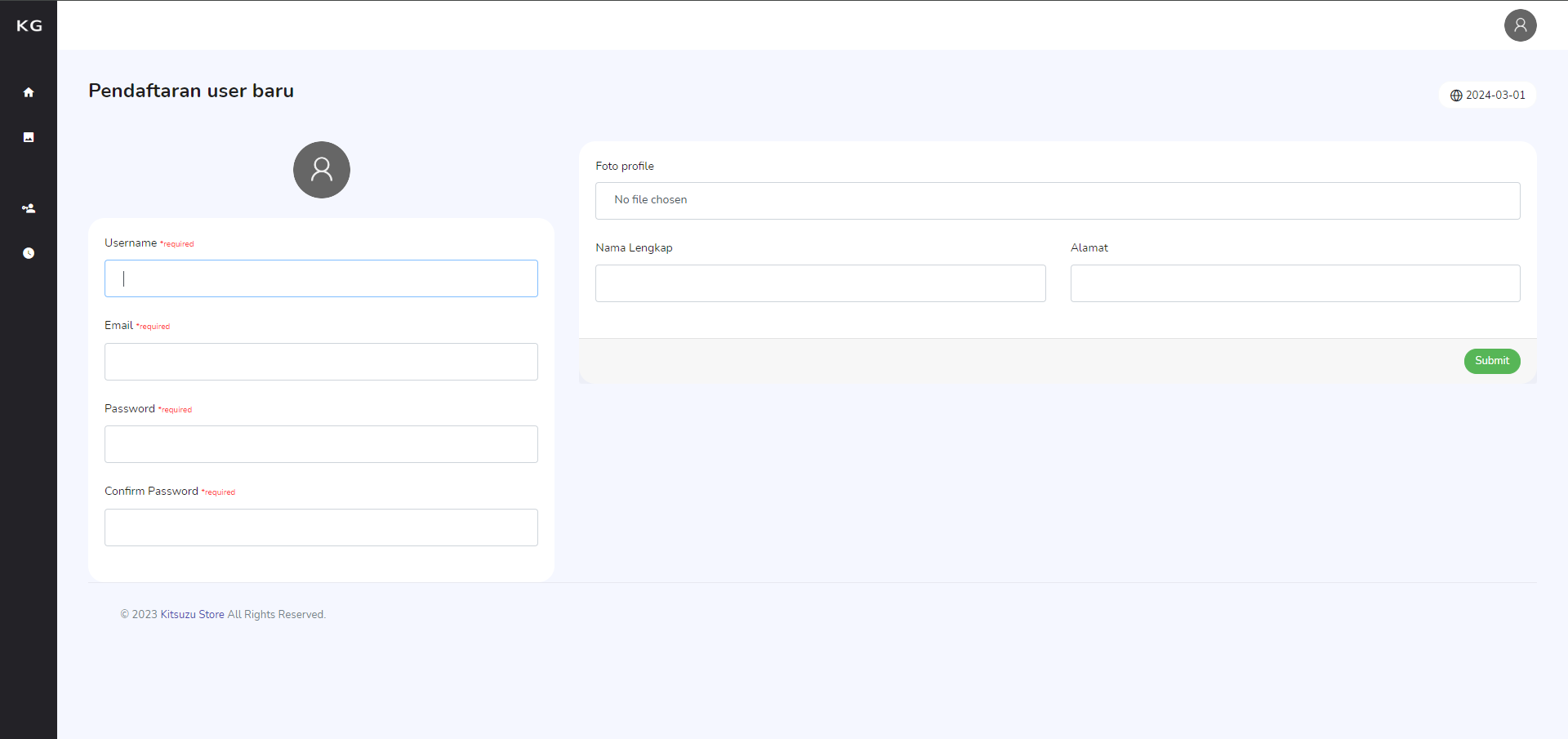
1. **Data User**

****

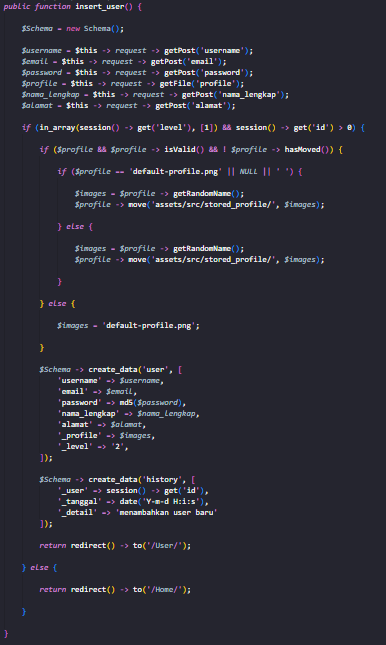
Pada halaman ini digunakan untuk menampilkan semua data user yang telah terdaftar pada sistem website gallery foto ini. Dimana untuk menampilkan datanya digunakan fungsi dari instance **Schema()** yaitu **visual\_table()** yang akan menampilkan data dari table user saja. Pada halaman ini terdapat data dari pengguna seperti **foto profil**, **username**, dan juga **email**.

****

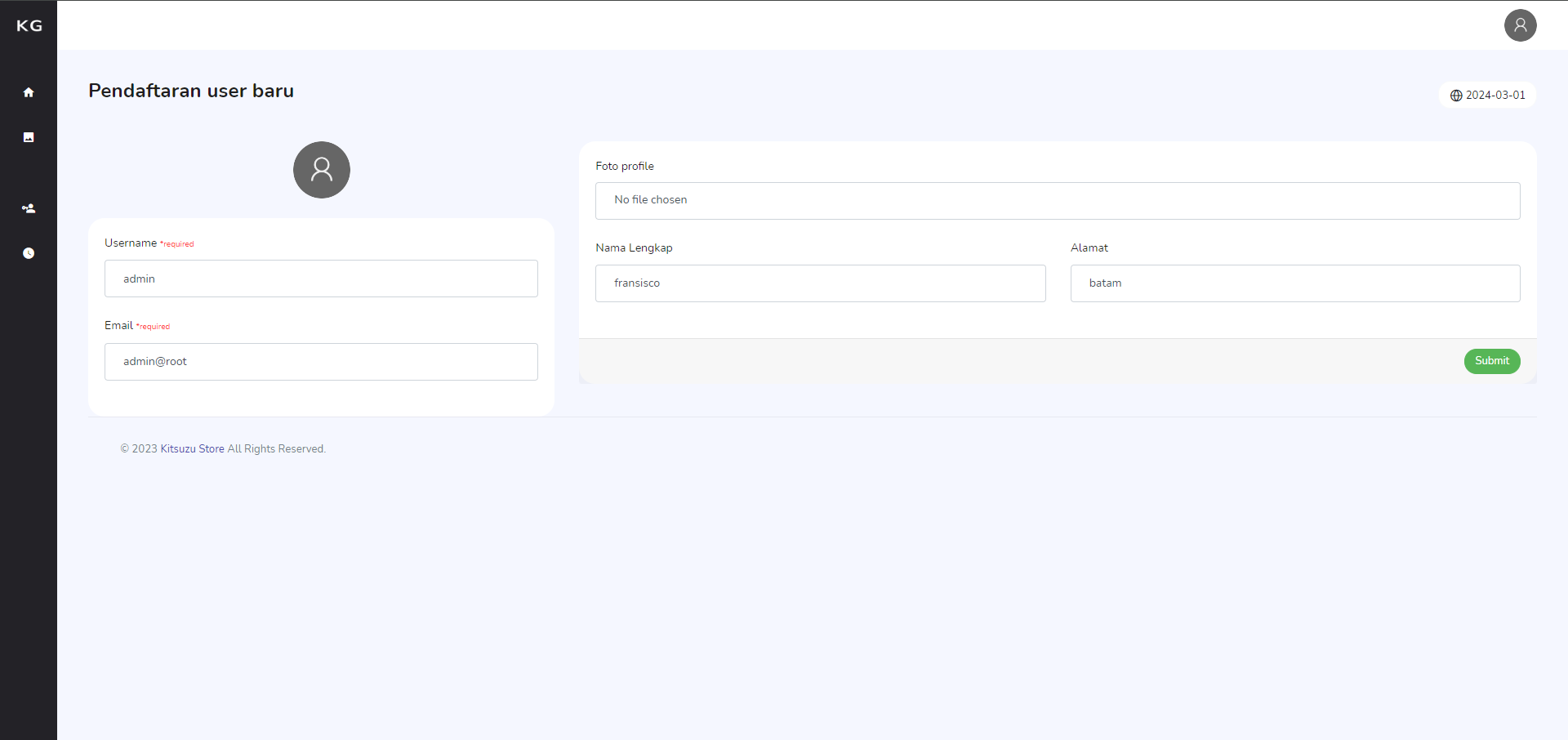
1. **Menambahkan User**

****

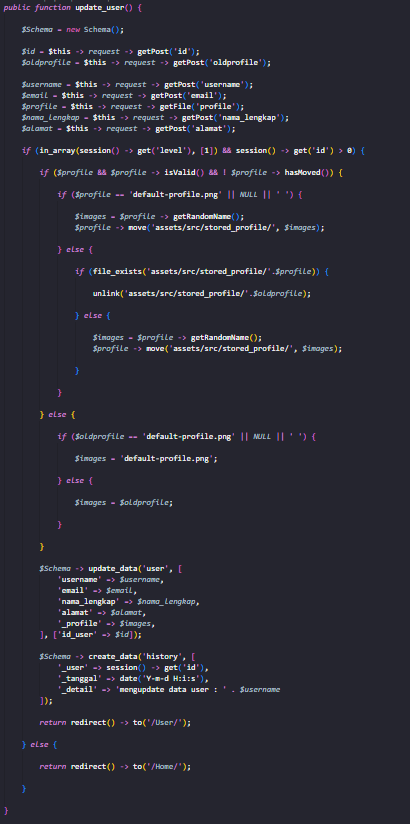
Pada controllernya terdapat fungsi **insert\_user()** yang dimana fungsi ini digunakan untuk menginput data user baru. Pada fungsi ini terdapat data yang di ambil yaitu berupa **username**, **email**, **password**, lalu user juga diwajibkan untuk melakukan konfirmasi password, **foto profil**, **nama lengkap**, dan juga **alamat** pengguna. lalu data tersebut dimasukkan ke dalam database menggunakan fungsi **create\_data()** yang tablenya di arahkan ke table user yang dimana akan diisi dengan data yang sudah dijadikan array. Lalu pada fungsi ini juga terdapat sistem penambahan history, yang dimana setiap kali pengguna membuat user baru maka akan di catat, dimana history ini mencakup **user**, **tanggal**, dan juga **detail** dari data yang dimasukkan.

****

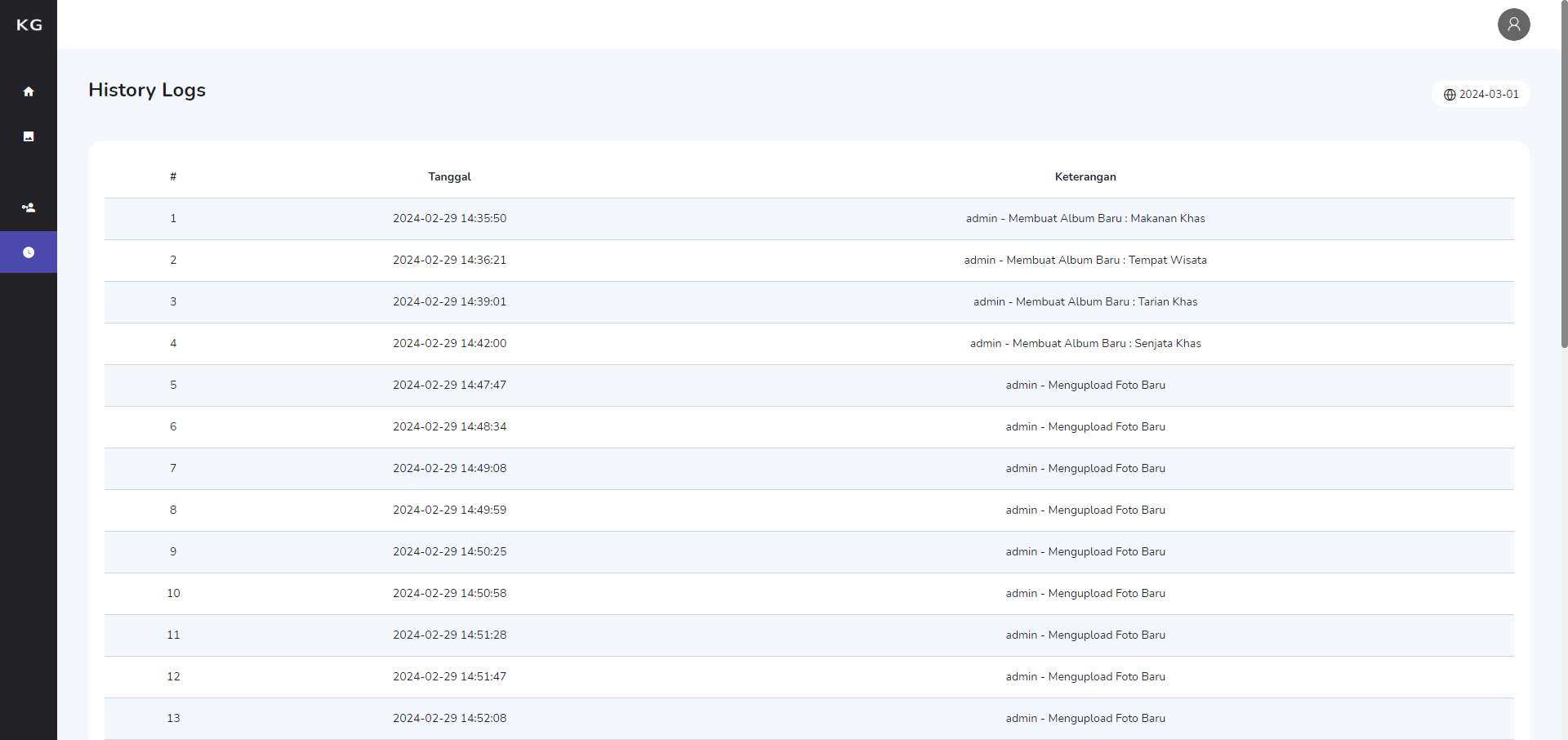
1. **Mengedit User**

****

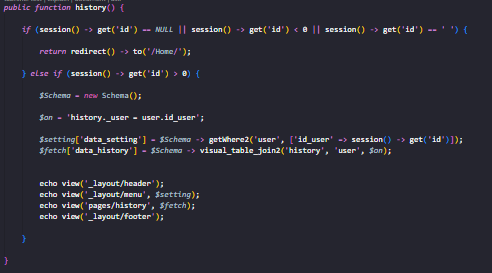
Pada halaman edit user ini menggunakan pemanggilan data menggunakan getWhere yang dimana datanya diperoleh sesuai dengan id\_user yang di pilih pada halaman user. Di halaman ini user dapat mengubah data yaitu seperti username, email, foto profile, nama lengkap, dan juga alamat. Yang lalu datanya akan di jadikan array pada controller fungsi **update\_user()** dan untuk dapat mengupdate data user ini diperlukan id\_usernya lagi dengan menggunakan input type hidden yang valuenya merupakan id\_user. Setelah itu semua datanya akan diperbaharui dengan menggunakan fungsi **update\_data()** dengan nama table sebagai **user** dan data yang akan diubah sebagai array, beserta id\_user yang akan digunakan untuk mencari data pada database yang akan diubah. Pada halaman ini juga terdapat validasi untuk foto profile yang akan di upload pada storage assets website ini, beserta mencatat history perubahan yang dilakukan untuk data **user**.

****

1. **History**

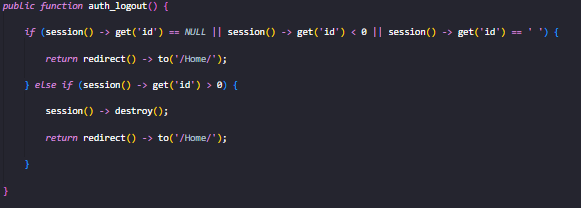
****

Pada halaman history ini hanya akun **admin** yang dapat mengaksesnya, semua data history ini akan secara otomatis ditambahkan setiap kali pengguna menambahkan **album** dan **foto** serta perubahan data yang terjadi kepada **album**, dan juga **foto**.

****

Pada controller ini terdapat fungsi **history()** yang dimana digunakan untuk menampilkan halaman history yang hanya bisa di akses oleh akun admin. Dan juga terdapat instansi **Schema()** yang digunakan untuk memanggil fungsi **getWhere2()** beserta **visual\_table\_join2()** yang digunakan untuk menampilkan data berdasarkan 2 table menggunakan join dari table history dan user, berdasarkan **history.\_user** = **user.id\_user**

1. **Logout**

****

Pada controllernya terdapat fungsi **auth\_logout()** yang dimana fungsi ini digunakan untuk melakukan proses logout pengguna pada website ini, fungsi ini terdapat validasi session dimana jika pengguna masih memiliki session maka session tersebut akan di hancurkan sehingga pengguna akan keluar dari halaman webshite gallery foto, dan pengguna otomatis akan kembali ke halaman login.