Nama : Ricky

NIM : 2602141101

Kelas : LB01

**FINAL ASSIGNMENT WEB PROGRAMMING**

**Link Github**: <https://github.com/Ricky1Z/JobFriends.git>

**Main Features:**

1. **[LO1, LO2,LO3 & LO4, 20 points]** User Authentication

* Implement the user authentication feature with login and registration ✅
* Perform validation according to the above conditions ✅

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **[LO1, LO2, LO3 & LO4, 10 points]** Payment

* Every user who registers is required to make a payment. ✅
* Payment using the Coin system ✅
* Conduct validation in accordance with the provisions above ✅

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **[LO1, LO2, LO3 & LO4, 30 points**] Friends

* Implement the friends feature, allowing users to add and remove friends by giving likes according to the specified topic. ✅
* Display the user's friend list on the profile. ✅

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **[LO1, LO2, LO3 & LO4, 20 points]** Messaging/Video Call

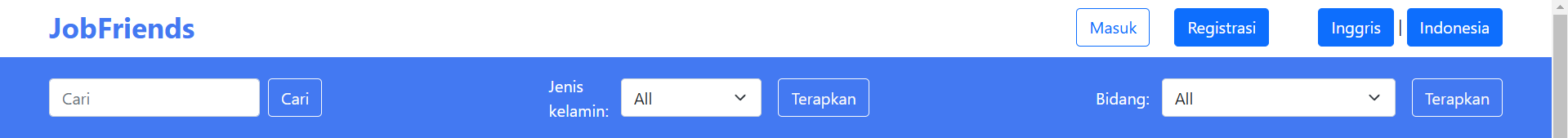
* Each user who has made friends, can communicate with each other through messages/video calls. ✅

A screenshot of a computer

Description automatically generated

1. **[LO1, LO2, LO3 & LO4, 10 points]** Searching & Filtering

* Implement searching & filtering features to search/filter by gender. Or search/filter by hobby/job. ✅



1. **[LO1, LO2, LO3 & LO4,10 points]** Notification

* Notify users of friend requests and incoming messages. ✅

A blue and white computer screen

Description automatically generated

**Additional Features [Optional] (You are free to choose additional features below):**

1. **[LO1, LO2, LO3 & LO4, 30 points]** Buy Avatar

* Users can buy an avatar ✅ and use the avatar as a profile picture. ❌
* The system must be able to deduct the total coins for each avatar purchase. ✅

A blue and white computer screen

Description automatically generated

Notes: User hanya bisa membeli avatar dan ditampilkan di profile page, namun tidak dapat digunakan untuk foto profil.

1. **[LO1, LO2, LO3 & LO4, 15 points]** Topup Coins

* Users can add 100 coins from each press of 1 button. ✅

A blue and white computer screen

Description automatically generated

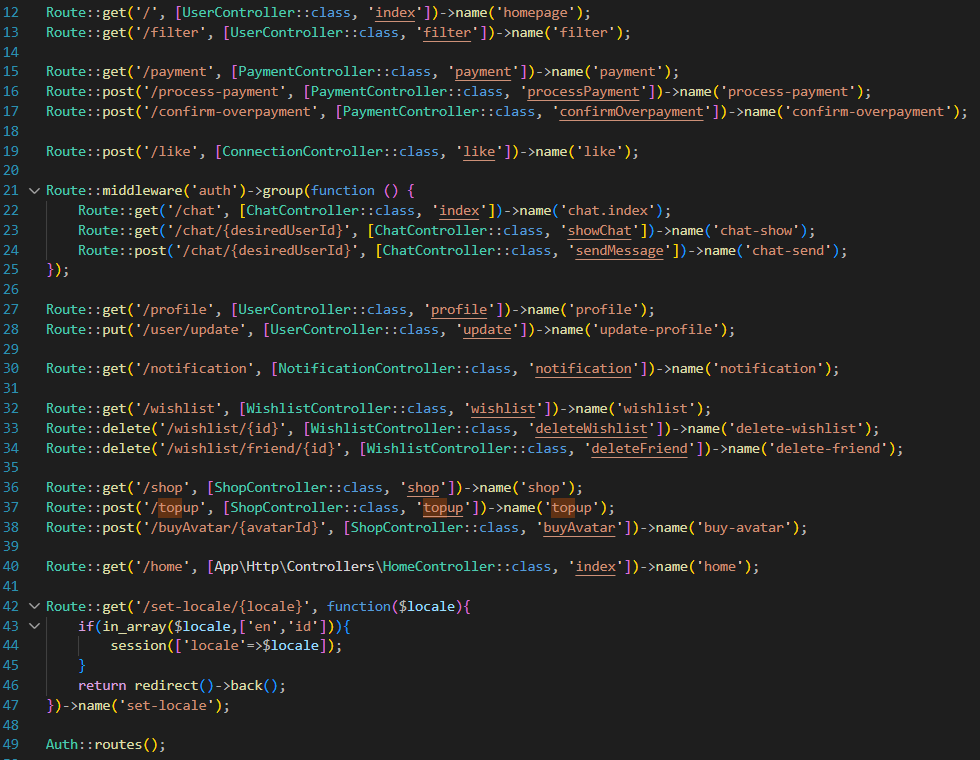
1. **[LO1, LO2, LO3 & LO4, 25 points]** Advanced Chat

* Users can send avatars to other users. ❌
* User can save avatars received from other users. ❌

1. **[LO1, LO2, LO3 & LO4, 30 points]** Visible Profile

* User can set profile visibility in the menu, so that it cannot be seen and cannot be searched by other users. ❌
* The system must be able to cut coins if using this feature. ❌
* The system must be able to change the profile photo to a bear image if using the invisible feature. ❌

**Route**



Disini terdapat 20 route:

1. '/'

* Route ini memanggil function index di UserController dan diberi nama “homepage”. Route ini untuk mengarahkan pengguna ke halaman utama (homepage).

1. '/filter'

* Route ini memanggil function filter di UserController dan diberi nama “filter”. Route ini untuk memproses filter sesuai dengan keinginan pengguna, dan mengarahkan pengguna balik ke halaman utama (homepage).

1. '/payment'

* Route ini memanggil function payment di PaymentController dan diberi nama “payment”. Route ini untuk mengarahkan pengguna ke halaman pembayaran.

1. '/process-payment'

* Route ini memanggil function processPayment di PaymentController dan diberi nama “process-payment”. Route ini untuk memproses data pembayaran yang dikirimkan dari form pembayaran.

1. '/confirm-overpayment'

* Route ini memanggil function confirmOverpayment di PaymentController dan diberi nama “confirm-overpayment”. Route ini untuk mengonfirmasi pembayaran berlebih atau kelebihan bayar.

1. '/like'

* Route ini memanggil function like di ConnectionController dan diberi nama “like”. Route ini untuk menangani aksi "like" yang dilakukan oleh pengguna (untuk fitur wishlist).

1. '/chat'

* Route ini memanggil function index di ChatController dan diberi nama “chat-index”. Route ini untuk menampilkan daftar teman pengguna.

1. '/chat/{desiredUserId}' (get)

* Route ini memanggil function showChat di ChatController dan diberi nama “chat-show”. Route ini untuk menampilkan isi percakapan dengan pengguna tertentu berdasarkan desiredUserId.

1. '/chat/{desiredUserId}' (post)

* Route ini memanggil function sendMessage di ChatController dan diberi nama “chat-send”. Route ini untuk mengirim pesan ke pengguna tertentu dalam percakapan.

1. '/profile'

* Route ini memanggil function profile di UserController dan diberi nama “profile”. Route ini untuk menampilkan halaman profil pengguna.

1. '/user/update'

* Route ini memanggil function update di UserController dan diberi nama “update-profile”. Route ini untuk memproses pembaruan data profil pengguna.

1. '/notification'

* Route ini memanggil function notification di NotificationController dan diberi nama “notification”. Route ini untuk menampilkan daftar notifikasi pengguna.

1. '/wishlist'

* Route ini memanggil function wishlist di WishlistController dan diberi nama “wishlist”. Route ini untuk menampilkan daftar wishlist dan teman pengguna.

1. '/wishlist/{id}'

* Route ini memanggil function deleteWishlist di WishlistController dan diberi nama “delete-wishlist”. Route ini untuk menghapus pengguna lain dari wishlist berdasarkan id.

1. '/wishlist/friend/{id}'

* Route ini memanggil function deleteFriend di WishlistController dan diberi nama “delete-friend”. Route ini untuk menghapus pengguna lain yang merupakan “teman” berdasarkan id.

1. '/shop'

* Route ini memanggil function shop di ShopController dan diberi nama “shop”. Route ini untuk menampilkan halaman belanja pengguna.

1. '/topup'

* Route ini memanggil function topup di ShopController dan diberi nama “topup”. Route ini untuk memproses top-up coin pengguna.

1. '/buyAvatar/{avatarId}'

* Route ini memanggil function buyAvatar di ShopController dan diberi nama “buy-avatar”. Route ini untuk membeli avatar tertentu berdasarkan avatarId.

1. '/home'

* Route ini memanggil function index di HomeController dan diberi nama “home”. Route ini untuk mengarahkan pengguna ke halaman utama setelah login.

1. '/set-locale/{locale}'

* Route ini diberi nama “set-locale” dan berfungsi untuk mengubah bahasa website berdasarkan parameter locale.

**Middleware**



Disini hanya terdapat 1 middleware:

1. SetLocale

* Middleware ini memastikan bahwa semua permintaan ke aplikasi menggunakan bahasa yang sesuai dengan preferensi pengguna (disimpan di sesi).

**Controller**

A screenshot of a computer

Description automatically generated

Disini terdapat 14 controller:

1. ConfirmPasswordController

* Mengelola konfirmasi ulang kata sandi pengguna saat diperlukan, seperti untuk tindakan sensitif (misalnya, mengubah pengaturan keamanan).
* Menggunakan trait ConfirmsPasswords.
* Mengarahkan pengguna ke /home jika URL yang dimaksudkan gagal.
* Menggunakan auth, sehingga hanya pengguna yang sudah login dapat mengakses fungsi konfirmasi ulang kata sandi.

1. ForgotPasswordController

* Mengelola pengiriman email untuk mengatur ulang kata sandi.
* Menggunakan trait SendsPasswordResetEmails untuk mengirim email reset password ke pengguna.
* Trait menyediakan logika bawaan untuk mengirimkan email dengan tautan reset password.

1. LoginController

* Mengelola proses login dan logout pengguna.
* Menggunakan trait AuthenticatesUsers.
* Mengarahkan pengguna ke /home, setelah berhasil login
* Middleware:
  + guest: Membatasi akses bagi pengguna yang belum login kecuali untuk logout.
  + auth: Membatasi fungsi logout hanya untuk pengguna yang login.

1. RegisterController

* Mengelola proses pendaftaran pengguna baru.
* Menggunakan trait RegistersUsers.
* Mengarahkan pengguna ke /home, setelah berhasil registrasi
* Method validator(array $data):
  + Memvalidasi input pendaftaran, termasuk nama, email, password, gender, dan sebagainya.
  + Validasi mendetail dilakukan menggunakan aturan seperti regex dan array checks.
* Method create(array $data):
  + Membuat pengguna baru di database dengan data yang sudah divalidasi.
  + Password dienkripsi menggunakan Hash::make().

1. ResetPasswordController

* Mengelola permintaan pengaturan ulang kata sandi pengguna.
* Menggunakan trait ResetsPasswords.
* Mengarahkan pengguna ke /home, setelah berhasil reset password
* Trait menyediakan logika untuk memverifikasi token reset, memperbarui kata sandi, dan login pengguna setelah reset.

1. VerificationController

* Mengelola verifikasi email untuk pengguna baru.
* Menggunakan trait VerifiesEmails.
* Mengarahkan pengguna ke /home, setelah berhasil verifikasi
* Middleware:
  + auth: Membatasi akses verifikasi hanya untuk pengguna yang login.
  + signed: Memastikan URL verifikasi sudah ditandatangani untuk keamanan.
  + throttle:6,1: Membatasi percobaan verifikasi ulang (maksimal 6 kali per menit).

1. ChatController

* Method index()
  + Menampilkan halaman utama chat yang berisi daftar pengguna yang terhubung (status connected).
* Method showChat($desiredUserId)
  + Menampilkan percakapan antara pengguna yang sedang login dan pengguna lain berdasarkan desiredUserId.
* Method sendMessage(Request $request, $desiredUserId)
  + Memproses pesan yang dikirimkan oleh pengguna yang sedang login.

1. ConnectionController

* Method like(Request $request)
  + Memproses permintaan "like" dari pengguna dan menentukan apakah hubungan pengguna akan tetap dalam status wishlist atau diubah menjadi status connected jika kedua pengguna saling menyukai.

1. HomeController

* Method \_\_construct()
  + Menerapkan middleware auth untuk memastikan hanya pengguna yang telah login yang dapat mengakses halaman ini.
* Method index()
  + Menampilkan halaman utama aplikasi (home).

1. NotificationController

* Method notification()
  + Mengambil semua notifikasi milik pengguna yang sedang login (berdasarkan Auth::id()) dan menampilkan pada halaman notification.

1. PaymentController

* Method payment()
  + Menampilkan halaman payment.
* Method processPayment(Request $request)
  + Memproses pembayaran, menangani kekurangan atau kelebihan pembayaran, dan memberikan koin kepada pengguna berdasarkan kondisi pembayaran.
* Method confirmOverpayment(Request $request)
  + Mengonfirmasi kelebihan pembayaran untuk dikonversi menjadi koin tambahan pengguna.

1. ShopController

* Method shop()
  + Menampilkan avatar yang belum dimiliki pengguna untuk ditampilkan pada halaman shop.
* Method topup()
  + Menambahkan koin (100 koin) ke saldo pengguna saat top-up dilakukan.
* Method buyAvatar($avatarId)
  + Memproses pembelian avatar, yaitu dengan memastikan apakah pengguna memiliki cukup koin dan memastikan avatar belum dimiliki pengguna. Jika berhasil, tambahkan avatar ke pengguna dan kurangi saldo koin pengguna.

1. UserController

* Method index(Request $request)
  + Menampilkan daftar pengguna berdasarkan pencarian (nama, profesi, skill, atau field).
* Method filter(Request $request)
  + Menyaring pengguna berdasarkan gender atau bidang pekerjaan (field).
* Method profile()
  + Menampilkan profil pengguna beserta avatar dan koneksi yang dimiliki.
* Method update(Request $request)
  + Memperbarui data pengguna berdasarkan input form dengan validasi yang ketat.

1. WishlistController

* Method wishlist()
  + Menampilkan daftar pengguna dalam status wishlist dan connected.
* Method deleteWishlist($id)
  + Menghapus koneksi wishlist dari daftar wishlist pengguna berdasarkan ID.
* Method deleteFriend($id)
  + Menghapus teman dari koneksi dan mengubah hubungan menjadi wishlist di sisi pengguna lain.

**Model**

A screenshot of a computer program

Description automatically generated

Disini terdapat 6 model:

1. Avatar

* Model Avatar mewakili entitas yang menyimpan informasi tentang avatar, seperti gambar avatar dan harga.
* Memiliki relasi Many-to-Many, di mana setiap avatar dapat dimiliki oleh banyak pengguna, dan setiap pengguna dapat memiliki banyak avatar. Relasi ini dijalankan melalui tabel pivot user\_avatars.
* Method users() untuk mendefinisikan relasi many-to-many antara model Avatar dan User.

1. Chat

* Model Chat digunakan untuk menyimpan percakapan antara dua pengguna. Tabel chats memiliki kolom user\_id, desired\_user\_id, dan chat, yang mencatat pesan antara dua pengguna.
* Memiliki relasi BelongsTo, di mana setiap chat terkait dengan dua pengguna (pengirim dan penerima). Relasi ini didefinisikan dengan fungsi user() dan desiredUser().
* Method:
  + Method user() untuk menghubungkan chat ke pengguna yang mengirim pesan.
  + Method desiredUser() untuk menghubungkan chat ke pengguna yang menerima pesan.

1. Connection

* Model Connection digunakan untuk menyimpan status koneksi antara dua pengguna. Setiap koneksi memiliki status, seperti "wishlist" atau "connected", yang menunjukkan apakah pengguna sedang berinteraksi atau tidak.
* Memiliki relasi BelongsTo, dimana relasi antara pengguna dan koneksi dijelaskan oleh user() dan desiredUser(), yang masing-masing merujuk pada pengguna yang terlibat dalam koneksi.
* Method:
  + Method user() untuk menghubungkan koneksi dengan pengguna yang membuat koneksi.
  + Method desiredUser() untuk menghubungkan koneksi dengan pengguna yang menjadi tujuan koneksi.

1. Notification

* Model Notification digunakan untuk menyimpan notifikasi yang diterima oleh pengguna. Notifikasi ini bisa berupa pemberitahuan tentang aktivitas tertentu, seperti koneksi atau pesan baru.
* Memiliki relasi BelongsTo, dimana setiap notifikasi terkait dengan satu pengguna. Fungsi user() menghubungkan notifikasi ke pengguna yang menerimanya.
* Method user() untuk menghubungkan notifikasi ke pengguna yang menerima pemberitahuan.

1. User

* Model User mewakili pengguna dalam aplikasi. Selain informasi dasar seperti nama dan email, model ini menyimpan data terkait profesi, keterampilan, avatar, dan status koin pengguna.
* Relasi:
  + Memiliki relasi HasMany, menghubungkan pengguna dengan koneksi mereka melalui connections().
  + Memiliki relasi BelongsToMany, menghubungkan pengguna dengan pengguna lain yang memiliki status "connected" melalui connectedUsers().
  + Memiliki relasi BelongsToMany, menghubungkan pengguna dengan avatar yang mereka miliki melalui avatars().
* Method:
  + Method connections() untuk mendefinisikan relasi satu ke banyak antara pengguna dan koneksi mereka.
  + Method connectedUsers() untuk mendefinisikan relasi banyak ke banyak antara pengguna yang terhubung (status "connected").
  + Method avatars() untuk mendefinisikan relasi banyak ke banyak antara pengguna dan avatar mereka.

1. UserAvatar

* Model UserAvatar adalah model pivot yang menghubungkan pengguna dan avatar mereka. Tabel ini menyimpan hubungan antara user\_id dan avatar\_id.
* BelongsTo: Relasi ini menghubungkan tabel pivot dengan model User dan Avatar.
* Method:
  + Method user() untuk menghubungkan entitas UserAvatar ke pengguna yang memiliki avatar.
  + Method avatar() untuk menghubungkan entitas UserAvatar ke avatar yang dimiliki oleh pengguna.

**Migration**

A screenshot of a computer program

Description automatically generated

Disini terdapat 6 migrations:

1. Users

* Tabel users menyimpan informasi tentang pengguna.
* Terdapat beberapa kolom seperti: name, email, password, gender, phone, linkedin, field, profession, skill, image, dan coin.

1. Avatars

* Tabel avatars menyimpan informasi tentang avatar.
* Terdapat beberapa kolom seperti: image dan price.

1. Connections

* Tabel connections menyimpan informasi tentang hubungan antar 2 pengguna.
* Terdapat beberapa kolom seperti: user\_id dan desired\_user\_id yang masing-masing merujuk ke pengguna yang terhubung satu sama lain. Dan kolom status untuk menyimpan status koneksi, seperti "wishlist" atau "connected".

1. User\_avatars

* Tabel user\_avatars adalah tabel pivot yang menghubungkan pengguna dengan avatar yang dimiliki.
* Terdapat beberapa kolom seperti: user\_id dan avatar\_id, yang masing-masing mengacu pada pengguna dan avatar.

1. Notifications

* Tabel notifications menyimpan informasi tentang notifikasi yang saling dikirimkan antar 2 pengguna.
* Terdapat beberapa kolom seperti: user\_id mengacu pada pengguna yang menerima pemberitahuan, dan kolom notification berisi isi pemberitahuan.

1. Chats

* Tabel chats menyimpan informasi tentang chat antar 2 pengguna.
* Terdapat beberapa kolom seperti: user\_id dan desired\_user\_id yang mengacu pada pengguna yang terlibat dalam percakapan (pengirim dan penerima).

**Seeder**

**A black background with white text

Description automatically generated**

Disini terdapat 1 seeder:

1. AvatarsSeeder

* Seeder ini bertujuan untuk mengisi tabel avatars dengan data avatar awal, yang dapat digunakan dalam aplikasi, misalnya sebagai pilihan avatar pengguna.

**View**

**A screenshot of a computer

Description automatically generated**

Disini terdapat 12 view:

1. Login.blade.php

A screenshot of a computer

Description automatically generated

* Halaman untuk memasukkan akun pengguna yang telah didaftarkan pada halaman registrasi.
* Terdapat label dan text field untuk email dan password; check box untuk remember me; tombol login; dan tombol forgot password.

1. Register.blade.php

A screenshot of a computer

Description automatically generated

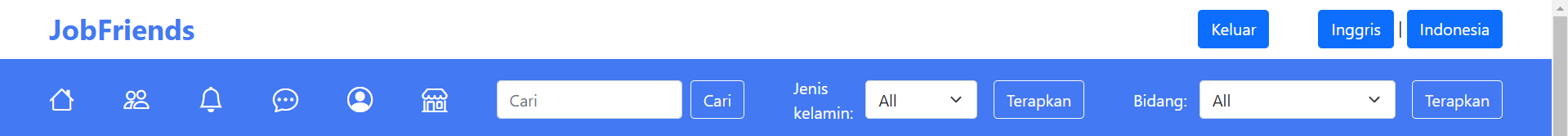
A screenshot of a computer

Description automatically generated

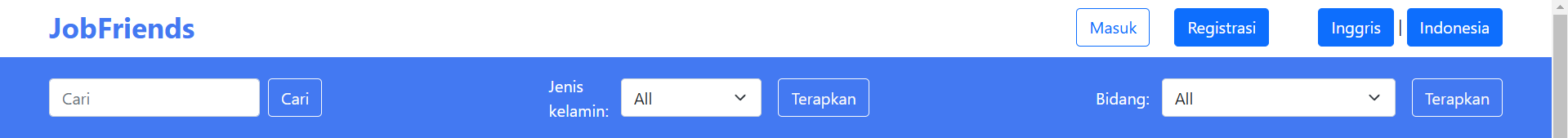
* Halaman untuk mendaftarkan akun pengguna agar bisa masuk ke halaman utama aplikasi.
* Terdapat label dan text field untuk name, phone, linkedin, profession, skill, email password, dan confirm password; radio button untuk pilih gender; check box untuk pilih field dan tombol registration.

1. Master.blade.php

* Sebagai template utama yang menyusun struktur dan tampilan aplikasi JobFriends.
* Berisi navbar yang terdiri dari nama aplikasi yaitu JobFriends; tombol logout, login, registration, ganti bahasa Indonesia atau inggris; icon homepage, wishlist dan friend, notification, chat, profile, dan shop; search bar; serta filter berdasarkan gender dan field.

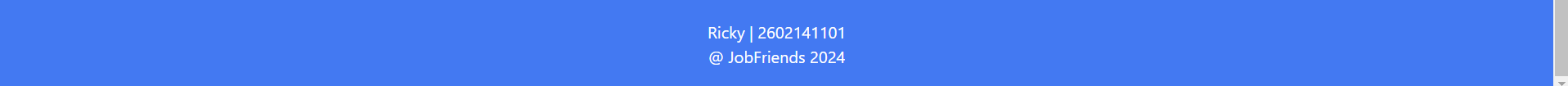


(Tampilan setelah login)



(Tampilan sebelum login)

* Selain itu, master page juga berisi footer yang menampilkan nama dan nim saya serta tahun pembuatan aplikasi JobFriends.



1. Chat.blade.php

A computer screen shot of a person

Description automatically generated

* Halaman ini untuk menampilkan daftar teman dari pengguna yang sedang login, ketika profil teman di klik, maka pengguna akan di direct ke halaman untuk chat.
* Terdapat nama page yaitu “Pesan”; tulisan “Daftar teman:” dan “(Pilih teman untuk ngobrol)”; serta profil teman dari pengguna yang sedang login.

1. Home.blade.php

A blue rectangle with white text

Description automatically generated

(Tampilan setelah login)

A blue and white box with text

Description automatically generated

(Tampilan sebelum login)

* Halaman ini merupakan halaman yang ditampilkan setelah pengguna melakukan registrasi atau login.
* Terdapat tombol “Bayar biaya masuk” apabila pengguna baru mendaftar diri, dan tombol homepage apabila pengguna sudah login.

1. Homepage.blade.php

A screenshot of a computer

Description automatically generated

* Halaman ini untuk menampilkan semua daftar pengguna yang telah mendaftarkan akunnya dan berisi data pribadi dari pengguna tersebut.
* Terdapat foto, nama, bidang yang dipilih, jenis kelamin, profesi, keahlian, dan linkedin dari setiap pengguna yang login; Serta tombol thumbs-up untuk memasukkan pengguna ke wishlist.

1. Notification.blade.php

A blue and white computer screen

Description automatically generated

* Halaman ini untuk menampilkan semua notifikasi yang diterima oleh pengguna yang sedang login.
* Terdapat nama halaman yaitu “Notifikasi”, serta semua teks notifikasi yang diterima oleh pengguna.

1. Payment.blade.php

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Halaman ini untuk melakukan pembayaran biaya registrasi bagi pengguna yang baru melakukan registrasi.
* Terdapat nama halaman yaitu “Pembayaran”; label dan text field untuk biaya registrasi; tombol submit payment; serta alert ketika pengguna overpaid dan underpaid.

1. Profile.blade.php

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A blue and white rectangle with a white arrow

Description automatically generated

* Halaman ini untuk menampilkan daftar teman, data pribadi, dan juga avatar yang dimiliki oleh pengguna. Selain itu pengguna juga dapat memperbarui data pribadi pengguna.
* Terdapat nama halaman yaitu “Profil”; foto dan nama dari teman pengguna; foto pengguna; jumlah koin yang dimiliki; label dan tempat untuk upload image; label dan text field untuk update data pengguna seperti nama, nomor telepon, linkedin, profesi, skill, dan email; radio button untuk update data gender; check box untuk update data field; tombol untuk update data; serta foto dari avatar yang dimiliki oleh pengguna.

1. SendMessage.blade.php

A screenshot of a computer

Description automatically generated

* Halaman ini untuk mengirim pesan antar pengguna dengan teman yang dipilih.
* Terdapat field untuk melihat semua pesan yang saling dikirimkan; field untuk input pesan yang ingin dikirim; serta tombol untuk mengirim teks.

1. Shop.blade.php

A blue and white computer screen

Description automatically generated

* Halaman ini untuk top-up coin dan juga membeli avatar.
* Terdapat nama halaman yaitu “Toko”; tombol untuk top-up 100 koin; jumlah koin pengguna; foto dan harga avatar yang masih tersedia; serta tombol untuk membeli avatar.

1. Wishlist.blade.php

A screenshot of a computer

Description automatically generated

* Halaman ini untuk menampilkan daftar keinginan dan daftar teman dari pengguna yang sedang login. Pengguna juga dapat menghapus pengguna lain dari daftar keinginan dan daftar teman.
* Terdapat judul “Daftar Keinginan” dan “Daftar Teman”; data diri dari pengguna lain seperti nama, bidang, profesi, keahlian, dan linkedin; serta tombol untuk menghapus pengguna lain.