**LAPORAN TUGAS**

**MATA KULIAH PENGUJIAN PERANGKAT LUNAK**



Disusun oleh:

Vanessa Aulia Syifa Yudiasmara (22091397084)

Anis Putri Purwanti (22091397099)

Danu Teguh Heri Prasetyo (22091397100)

Dosen Pengampu:

Andi Iwan Nurhidayat, S.Kom., M. T.

**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA**

**FAKULTAS VOKASI**

**UNIVERSITAS NEGERI SURABAYA**

**2024**

**Membuat Model “User”**



Model User ini memiliki atribut-atribut yang dapat diisi (fillable), yaitu 'username', 'password', dan 'name'. Atribut-atribut tersebut dapat diisi secara massal (mass assignment) menggunakan metode create() atau fill().

Beberapa atribut seperti 'password' dan 'remember\_token' disembunyikan (hidden) agar tidak disertakan ketika model User dikonversi menjadi array atau JSON, misalnya untuk tujuan respons API atau tampilan. Hal ini untuk keamanan data pengguna.

Model ini juga menggunakan trait HasFactory dan Notifiable. Trait HasFactory memberikan metode-factory untuk model, sementara trait Notifiable menambahkan kemampuan untuk mengirim notifikasi kepada pengguna.

**Membuat Model “Contact”**



Model Contact ini menggunakan trait HasFactory, yang memberikan kemampuan untuk menggunakan factory dalam membuat data palsu untuk model.

Model ini memiliki atribut-atribut yang dapat diisi (fillable), yaitu 'user\_id', 'first\_name', 'last\_name', 'email', dan 'phone'. Atribut-atribut tersebut dapat diisi secara massal (mass assignment) menggunakan metode create() atau fill().

Selain itu, model ini juga mendefinisikan relasi belongsTo dengan model User, yang menunjukkan bahwa setiap kontak dimiliki oleh satu pengguna (user). Artinya, model Contact memiliki metode user() yang mengembalikan relasi belongsTo dengan model User. Ini memungkinkan untuk mengakses data pengguna terkait dengan kontak tertentu.

**Membuat Model “Address”**



Model Address ini menggunakan trait HasFactory, yang memberikan kemampuan untuk menggunakan factory dalam membuat data palsu untuk model.

Model ini memiliki atribut-atribut yang dapat diisi (fillable), yaitu 'contact\_id', 'street', 'city', 'province', 'country', dan 'postal\_code'. Atribut-atribut tersebut dapat diisi secara massal (mass assignment) menggunakan metode create() atau fill().

Selain itu, model ini juga mendefinisikan relasi belongsTo dengan model Contact, yang menunjukkan bahwa setiap alamat dimiliki oleh satu kontak. Artinya, model Address memiliki metode contact() yang mengembalikan relasi belongsTo dengan model Contact. Ini memungkinkan untuk mengakses data kontak terkait dengan alamat tertentu.

**Membuat Migration “Users”**



Tabel ini akan memiliki kolom-kolom seperti id (primary key), username (unik), password, name, serta kolom timestamps untuk menandai waktu pembuatan dan pembaruan data. Metode up() digunakan untuk membuat tabel tersebut, sedangkan down() digunakan untuk menghapus tabel tersebut jika migration di-rollback. Tabel "users" ini biasanya digunakan untuk menyimpan informasi pengguna dalam aplikasi.

**Membuat Migration “Contacts”**



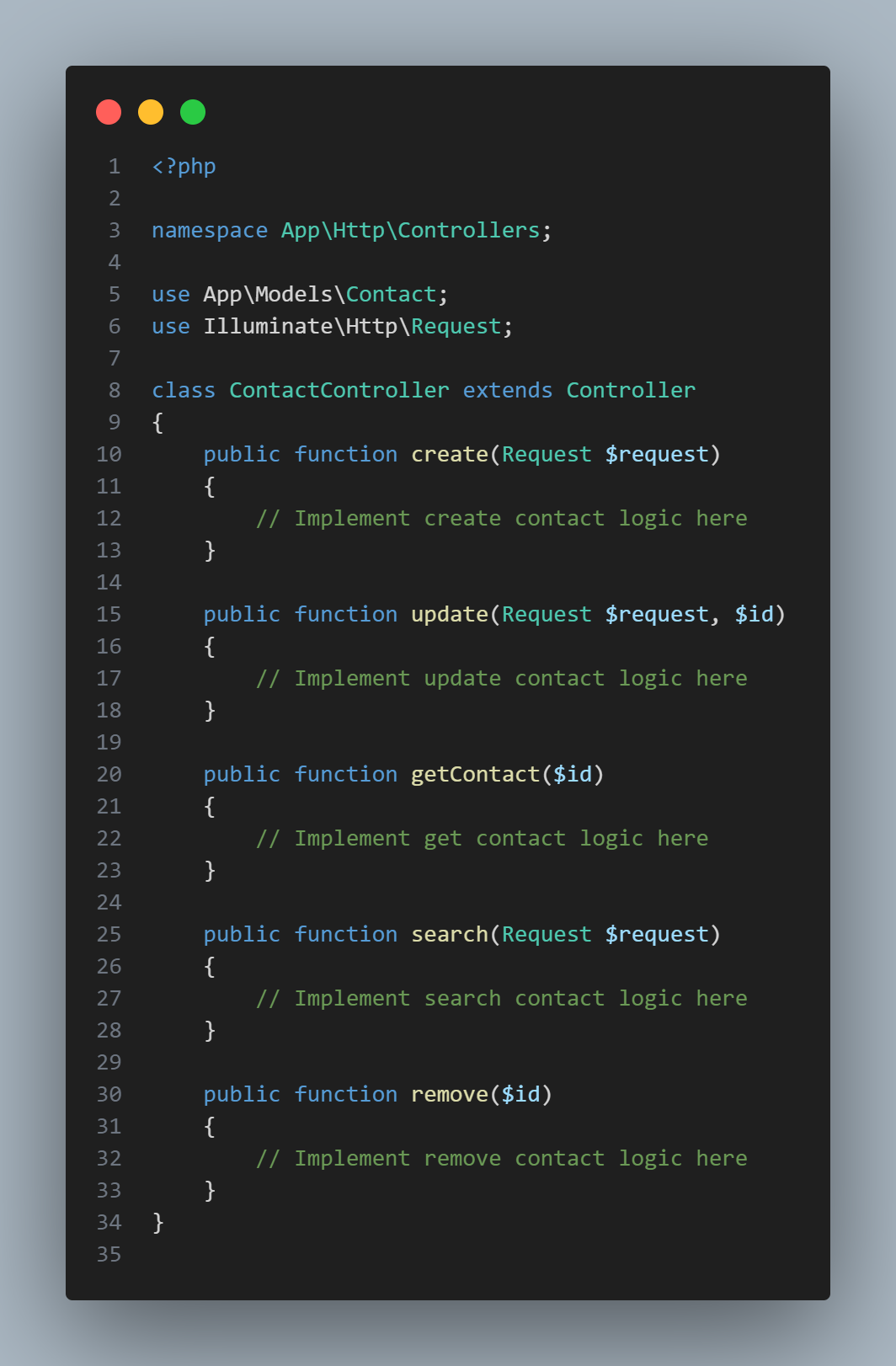
Tabel ini akan memiliki kolom-kolom seperti id (primary key), user\_id (foreign key yang terhubung ke tabel users), first\_name, last\_name, email, phone, serta kolom timestamps untuk menandai waktu pembuatan dan pembaruan data. Metode up() digunakan untuk membuat tabel tersebut, sedangkan down() digunakan untuk menghapus tabel tersebut jika migration di-rollback.

**Membuat Migration “Addresses”**



Migration ini bertujuan untuk membuat tabel "addresses" dalam database, dengan kolom-kolom yang sesuai. Tabel "addresses" akan memiliki kolom-kolom seperti id (primary key), contact\_id (foreign key yang terhubung ke tabel contacts), street, city, province, country, postal\_code, serta kolom timestamps untuk menandai waktu pembuatan dan pembaruan data. Metode up() digunakan untuk membuat tabel tersebut, sedangkan down() digunakan untuk menghapus tabel tersebut jika migration di-rollback.

**Membuat “Contact Controller”**



Program tersebut adalah bagian dari kontroler yang berfungsi untuk mengatur berbagai operasi terkait dengan entitas kontak, seperti membuat kontak baru, memperbarui kontak yang sudah ada, mengambil kontak berdasarkan ID, mencari kontak, dan menghapus kontak.

**Membuat “Address Controller”**



Program tersebut berfungsi untuk mengatur berbagai operasi terkait dengan entitas alamat, seperti membuat alamat baru, memperbarui alamat yang sudah ada, mengambil alamat berdasarkan ID, mencari alamat, dan menghapus alamat.

**Membuat “User Controller”**



Program tersebut berfungsi untuk mengelola pengguna (user) dan otentikasi dan bertanggung jawab untuk menangani operasi-operasi dasar yang terkait dengan pengelolaan pengguna, termasuk pendaftaran, login, logout, pembaruan, dan pengambilan informasi pengguna.

**Membuat Address Blade**



Kode tersebut merupakan bagian dari tampilan HTML untuk halaman web yang digunakan dalam sebuah proyek dengan judul "TUGAS PPL KELOMPOK 9". Halaman ini memiliki form untuk membuat alamat baru dengan mengirimkan data melalui metode POST ke endpoint "/addresses". Form tersebut memiliki input untuk memasukkan informasi tentang alamat, seperti jalan (street), kota (city), provinsi (province), negara (country), dan kode pos (postal code). Setelah semua informasi dimasukkan, pengguna dapat menekan tombol "Create Address" untuk membuat alamat baru dalam sistem.

**Membuat “Contact Controller”**



Tampilan HTML tersebut adalah bagian dari antarmuka pengguna untuk sistem manajemen kontak. Ini memungkinkan pengguna untuk berinteraksi dengan fitur-fitur manajemen kontak melalui browser web. Setiap form memiliki aksi yang berbeda dan berinteraksi dengan backend aplikasi melalui HTTP methods (POST atau GET) sesuai dengan kebutuhan operasi yang dilakukan, seperti membuat, memperbarui, mengambil, mencari, atau menghapus kontak.

**Membuat Index Blade**



Kode HTML di atas adalah bagian dari tampilan antarmuka pengguna untuk sebuah proyek yang disebut "TUGAS PPL KELOMPOK 9". Halaman ini menyediakan tautan menuju berbagai bagian manajemen dalam proyek tersebut, yang meliputi: judul halaman, user management section, contact management section, dan address management section. Setiap bagian memiliki tautan yang sesuai (href) yang mengarahkan pengguna ke halaman yang tepat dalam proyek, seperti halaman pendaftaran, manajemen kontak, atau manajemen alamat. Gaya CSS juga digunakan untuk menghapus garis bawah dari tombol-tombol tersebut agar tampilannya lebih rapi.

**Membuat Register Blade**



Kode HTML di atas adalah bagian dari tampilan antarmuka pengguna untuk halaman pendaftaran pengguna (user registration). Menampilkan judul halaman "User Registration", memberikan informasi tentang tujuan atau fungsi halaman tersebut dan Form Pendaftaran Pengguna yang memungkinkan pengguna untuk mendaftar dengan mengisi informasi yang diperlukan. Form ini mengirimkan data melalui metode POST ke endpoint "/register" yang kemungkinan besar akan diproses oleh backend untuk membuat akun baru.

Setelah formulir dikirim, data yang dimasukkan oleh pengguna akan dikirimkan ke server melalui metode POST, yang kemungkinan akan diproses oleh backend untuk membuat akun pengguna baru dalam sistem.

**Membuat Routes web.php**

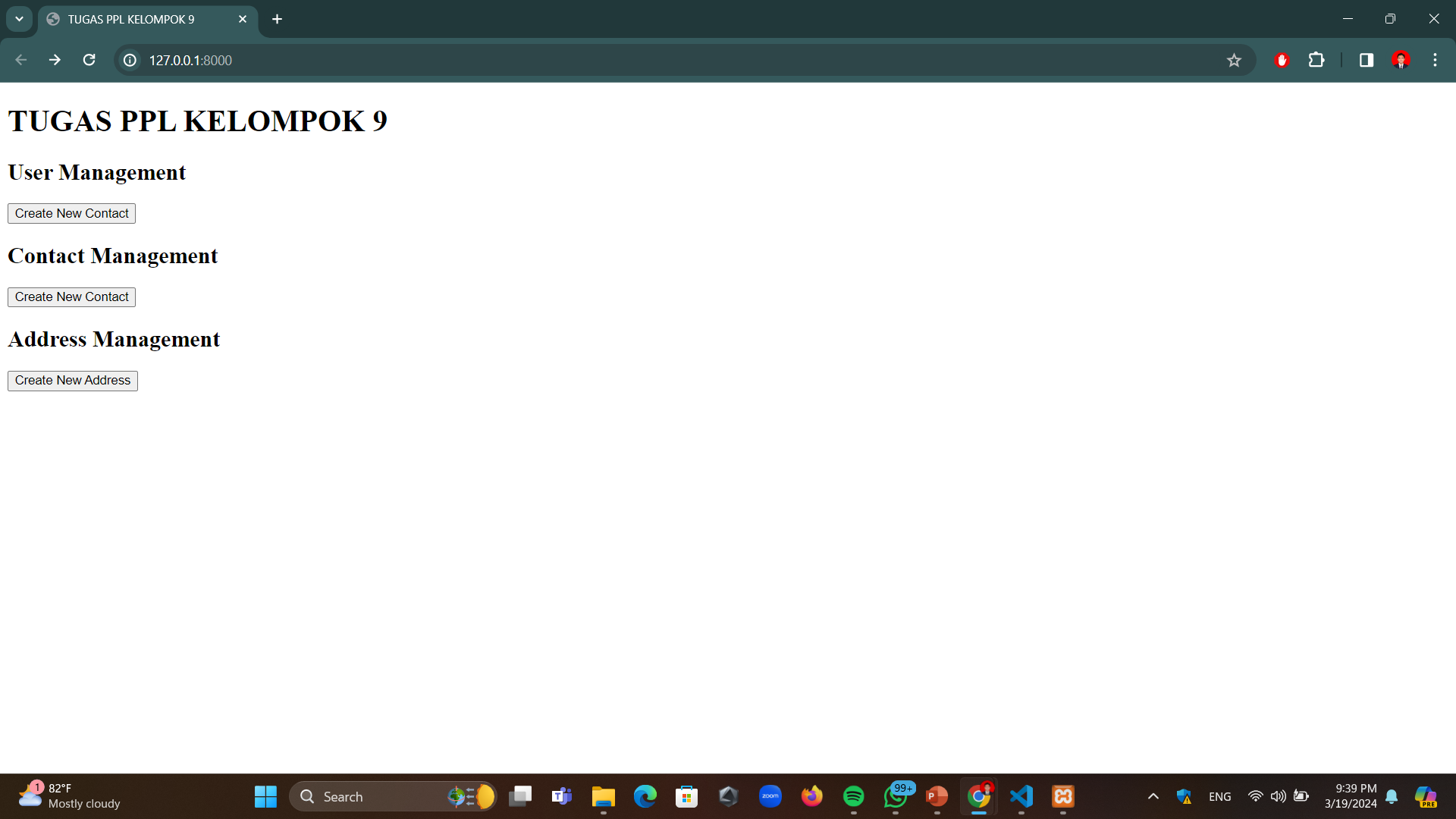


Program tersebut adalah file routes dalam framework Laravel yang mendefinisikan berbagai rute HTTP untuk aplikasi web. Setiap rute menghubungkan sebuah URL dengan sebuah fungsi atau metode pada sebuah controller.

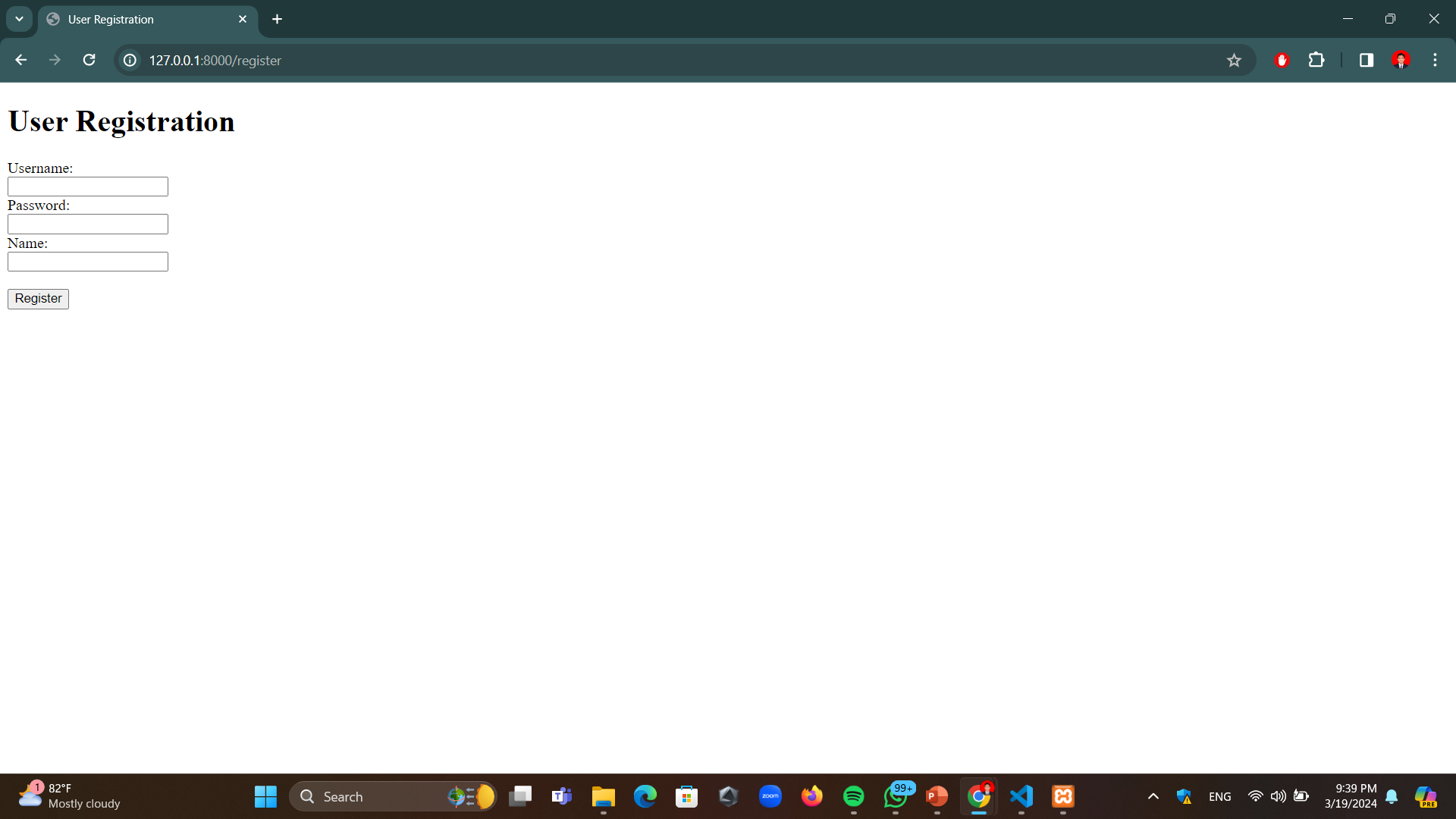
* Rute-rute yang diawali dengan /register, /login, /users, dan /logout terkait dengan operasi-operasi pengguna seperti registrasi, login, pengambilan data pengguna, pembaruan data pengguna, dan logout.
* Rute-rute yang diawali dengan /contacts terkait dengan operasi-operasi terkait kontak seperti pembuatan kontak, pembaruan kontak, pengambilan data kontak, pencarian kontak, dan penghapusan kontak.
* Rute-rute yang diawali dengan /addresses terkait dengan operasi-operasi terkait alamat seperti pembuatan alamat, pembaruan alamat, pengambilan data alamat, pencarian alamat, dan penghapusan alamat.

Rute-rute tersebut mengarahkan permintaan HTTP ke metode-metode yang sesuai pada controller UserController, ContactController, dan AddressController untuk mengelola data pengguna, kontak, dan alamat dalam aplikasi.

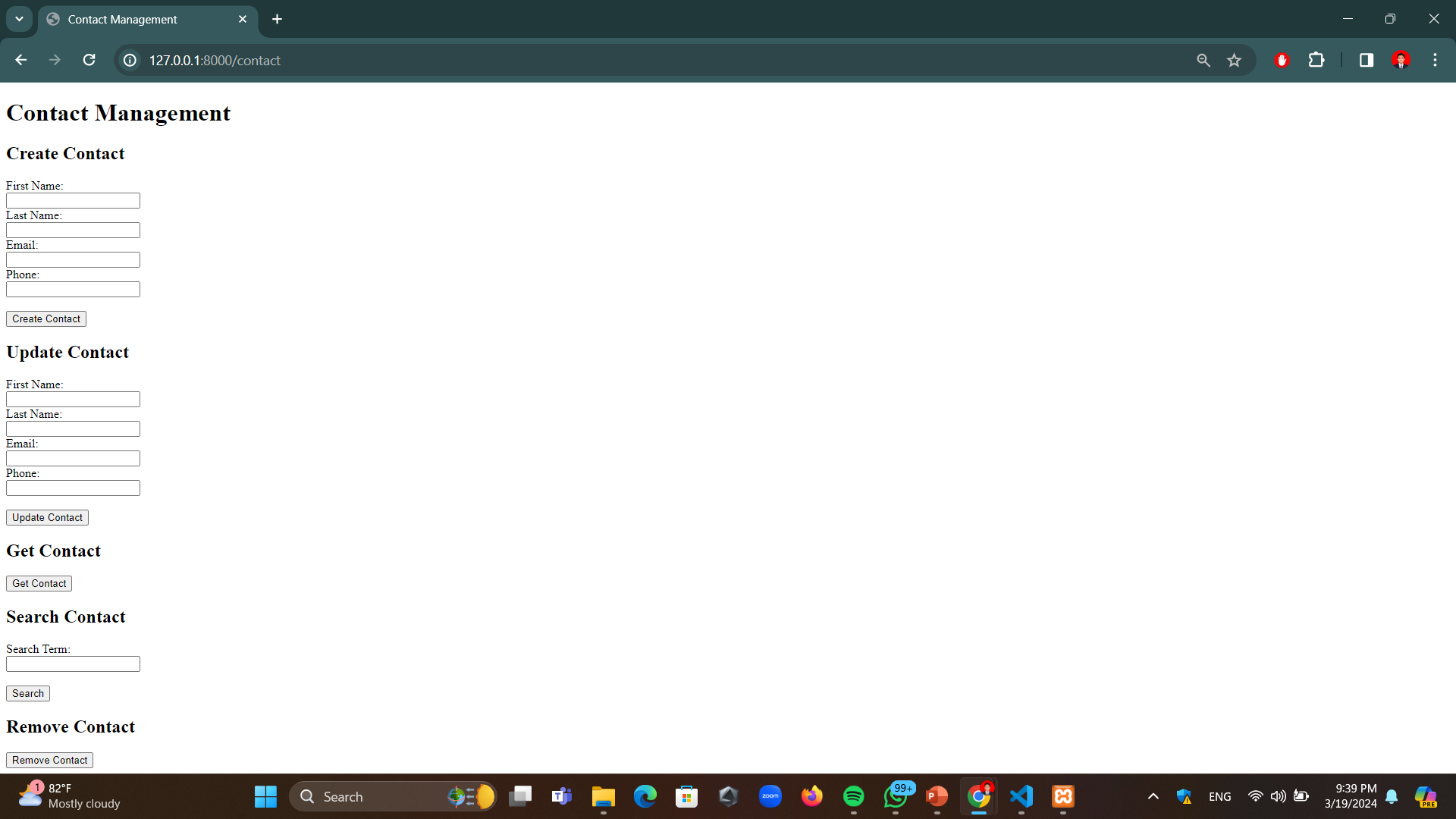
**Hasil:**



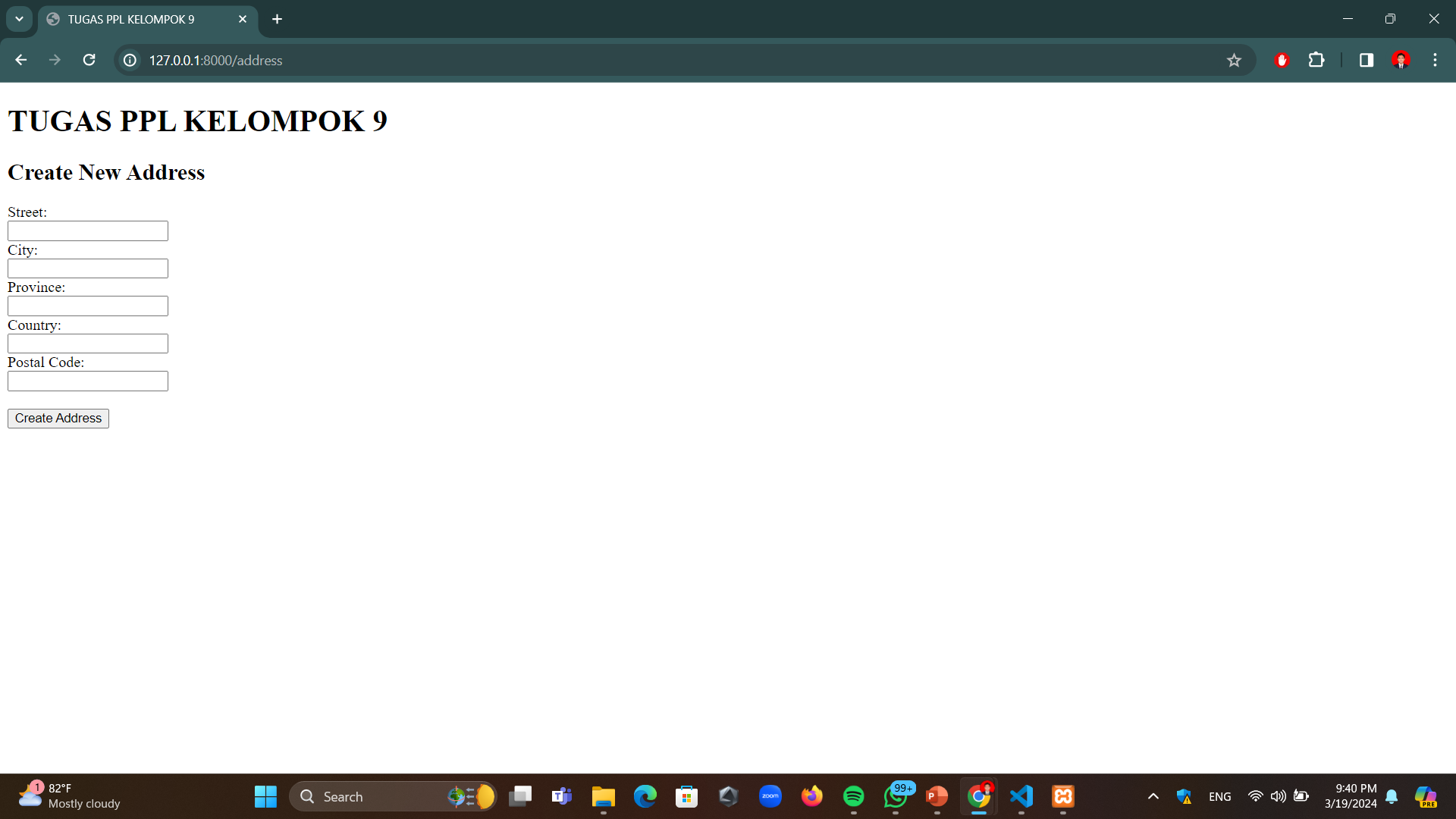
Gambar diatas adalah tampilan awal router 127.0.0.1.8000 yang berisi pilihan user, contact, dan address management.



Gambar diatas merupakan tampilan dari user registration hasil routing dari 127.0.0.1.8000/register.



Gambar diatas adalah tampilan dari Contact Management yang merupakan hasil routing dari 127.0.0.1.8000/contact.



Gambar diatas merupakan tampilan dari form create address yang berisi street, city, province, country, dan postal code yang merupakan hasil routing dari 127.0.0.1.8000/address.