1. Membuat halaman untuk menampilkan semua mahasiswa fasilkom

* Terdapat list yang berisi daftar mahasiswa fasilkom, yang dipanggil dari django model.

*Pada saat lab-7 dibuka, maka urls akan pemanggil fungsi index di views. Pada views, sebelumnya di inisiasi dulu class CSUIhelper() yang akan membuka data API cs ui. Method pertama yang dijalankan adalah mengambil akses token dengan username dan password dan client ID. Setelah mendapat token, memanggil method get\_mahasiswa\_list() untuk mengambil list daftar mahasiswa fasilkom. Method get\_mahasiswa\_list di csui helper hanya mengambil 100, dan yang di-load hanya mahasiswa dengan npm awal saja. Akan tetapi jika mau load yang npm ke belakang dapat diberi parameter method get\_mahasiswa( page ke berapa dari seluruh data di api).*

*Pada fungsi index, get\_mahasiswa\_list() daric sui helper dimasukkan ke dalam variable mahasiswa\_list. Friend\_list akan menyimpan objek Friend yang sudah dimasukkan ke dalam database (yang sudah ditambah sbg teman). List daftar mahasiswa fasilkom di return sebagai response dari fungsi index, ditampilkan lagi di lab\_7.html yang akan me-loop setiap mahasiswa yang ada dalam variable mahasiswa\_list di views, yang ditampilkan dalam bentuk nama dan npm mereka.*

* Buatlah tombol untuk dapat menambahkan list mahasiswa kedalam daftar teman (implementasikan menggunakan ajax).

*Dalam lab\_7.html, dibuat html yang membuat button yang pada saat di klik memanggil fungsi addFriend di javascript yang memiliki parameter nama dan npm dari mahasiswa\_list. Fungsi addFriend memanggil urls add-friend, yang kemudian memanggil fungsi add\_friend di views.*

* Mengimplentasikan validate\_npm untuk mengecek apakah teman yang ingin dimasukkan sudah ada didalam daftar teman atau belum.

*Diimplementasikan pada fungsi validate\_npm di mana objek friend di filter telah memiliki nilai npm yang sama dengan npm yang akan divalidasi sudah ada atau belum. Returnnya adalah true atau false. Method ini digunakan pada add friend, sehingga jika teman belum ada, maka akan ditambahkan. Tetapi jika teman sudah ada, tidak ditambah lagi.*

* Membuat pagination (hint: salah satu data yang didapat dari kembalian api.cs.ui.ac.id adalah next dan previous yang bisa digunakan dalam membuat pagination)

*Membuat pagination dimulai dari views.py, memanggil get variable page yang akan mengembalikan page yang sudah ad ajika variable telah diinisiasi sebelumnya, atau mengembalikan 1 jika belum. Llalu memanggil fungsi dari Paginator yang akan memngelompokkan mahasiswa 10 demi 10. Variable mahasiswa akan diinisasi halamannya, misalnya mahasiswa urutan ke 9 memiliki nomor halaman 1. Jika halaman yang akan diinisiasikan kepada mahasiswa bukan merupakan integer, maka mahasiswa akan diinisiasikan dengan nomor halaman 1. Apabila halaman kosong, maka mahasiswa akan diinisiasikan dengan nomor halaman saat itu. Index dimulai dari 0, tetapi mahasiswa.number memakai page (variable di baris 23) yang dimulai dari index ke-1, shg harus dikurangi 1 agar mulai dari 0. Start index dimulai dari 0, sedangkan end index adalah 10. Lalu membuat page range yang mencakup dari indeks mulai sampai indeks selesai. Pagination dikembalikan dalam bentuk respons yang akan dikembalikan ke lab\_7.html. Pada lab\_7.html, jika mahasiswa\_list yang sudah diberi nomor, memiliki page lain, maka akan membuat pagination dengan memanggil url mahasiswa dengan nomor sebelumnya atau sesudahnya (dengan metode a href yang akan memanggil url ?page={{page}}.*

1. Membuat halaman untuk menampilkan daftar teman

* Terdapat list yang berisi daftar teman, data daftar teman didapat menggunakan ajax.

*Pada daftar\_teman.html, jika document ready, maka akan di GET oleh ajax, semua friend yg ada di friend\_list. FriendList akan di parse menjadi bentuk JSON, lalu di append ke dalam id friend-list. Fungsi di views.py yaitu friend\_list dari objek Friend yang merupakan nama mahasiswa yang sudah di add. Lalu memanggil ke lab\_7/daftar\_teman.html lalu dirender.*

* Buatlah tombol untuk dapat menghapus teman dari daftar teman (implementasikan menggunakan ajax).

*Membuat tombol lain yaitu tombol delete pada daftar\_teman.html, yang apabila di klik akan menjalankan fungsi deleteFriend di bawahnya yang memanfaatkan id friend. Dari fungsi deleteFriend akan dipanggil urls /delete-friend/ dengan parameter id yang akan memanggil fungsi delete\_friend di views. Tombol delete diperbaiki dengan menambahkan menjadi table, sehingga tombol delete dapat diletakkan di sebelah nama orang.*

1. Pastikan kalian memiliki Code Coverage yang baik

* Jika kalian belum melakukan konfigurasi untuk menampilkan Code Coverage di Gitlab maka lihat langkah Show Code Coverage in Gitlab di README.md
* Pastikan Code Coverage kalian 100%