

BAB 3

PELAKSANAAN KERJA MAGANG

3.1. Kedudukan dan Koordinasi

Praktek kerja magang di PT. W3O Indo Scientia memegang jabatan *Web Developer Intern* yang bekerja bersama dengan tim *Web Developer* proyek ini, sesuai dengan Gambar 2.2. Koordinasi dan pengawasan oleh Bapak Aditya Bharata selaku *Senior System Analyst* dan dibimbing oleh Bapak Andrey selaku *Senior Web Developer*. Dalam proyek ini juga ada satu *developer* yang ikut membantu membangun aplikasi ini. Pemberian tugas diberikan oleh Bapak Hoi Chong yang disampaikan kepada Bapak Andrey lalu disampaikan ke penulis.

Segala pertanyaan, pembahasan dan hasil *bug* diinformasikan melalui aplikasi *chat Skype*, setelah selesai mengerjakan bagian yang ditugaskan, pada akhir hari hasil kode dikumpul kepada Bapak Andrey untuk diperiksa dan digabungkan ke server utama dan jika ada *update* dari server utama akan dikirim juga melalui via *Skype*.

3.2. Tugas yang Dilakukan

Selama praktek kerja magang ini, penulis diberi tugas sebagai berikut:

- Latihan mempelajari dan menganalisis *framework Ember.js*.
- Latihan mempelajari dan mengenal *framework Sails.js*.
- Membuat *database structure* dengan *Database PostgreSQL*.

- Membuat halaman *front-end website* dengan *Ember.js*.
- Membuat *back-end API website* dengan *Sails.js*.

3.3. Uraian Pelaksanaan Kerja Magang

Praktek kerja magang dilakukan selama 58 hari atau 8 minggu 6 hari, dimulai dari tanggal 24 Februari 2021 sampai 22 April 2021. Dapat dijabarkan sesuai dengan Tabel 3.1 berikut.

Tabel 3.1 Jadwal Kerja Magang

Minggu	Kegiatan
1	<ul style="list-style-type: none"> • Mempelajari <i>Sails.js</i> • Mempelajari <i>Ember.js</i> • Membuat <i>database</i> pada <i>PostgreSQL</i>
2	<ul style="list-style-type: none"> • Membuat halaman <i>home page</i> • Membuat halaman <i>Login</i> • Membuat halaman <i>Sign Up</i>
3	<ul style="list-style-type: none"> • Membuat halaman <i>New Client</i> • Membuat halaman <i>Generate Video</i> dan <i>Choose Video</i> • Membuat halaman <i>Subscription</i>
4	<ul style="list-style-type: none"> • Membuat halaman <i>Profile</i> • Membuat halaman <i>Report</i>
5	<ul style="list-style-type: none"> • Membuat <i>back-end homepage</i>
6	<ul style="list-style-type: none"> • Membuat <i>back-end Login</i> • Membuat <i>back-end Sign-up</i>
7	<ul style="list-style-type: none"> • Membuat <i>back-end Profile</i> • Membuat <i>back-end NewClient</i>
8	<ul style="list-style-type: none"> • <i>Cleaning</i> dan <i>Bug-fix</i>

3.3.1. Framework yang Digunakan

Dalam perancangan *website* ini *framework front-end* yang digunakan adalah *Ember.js*, kerangka kerja yang menggunakan metode MVC (model-view-controller) dengan bahasa *handlebar* untuk *view*-nya yang bekerja hampir sama dengan HTML namun memiliki fungsi tambahan seperti *component* untuk mempermudah *refactoring*, ada *action* juga yang bekerja sama seperti *function* dengan *controller*; untuk *model* dan *controller* digunakan bahasa *javascript* yang dihubungkan dengan *database* PouchDB, sebagai *database* yang bersifat *local-based*, dalam pengembangan *website* ini tidak memiliki model.

Framework *back-end* yang digunakan adalah *Sails.js* yang merupakan kerangka kerja yang menggunakan bahasa *javascript* yang memiliki metode MVC juga, *Sails.js* memiliki fitur tambahan seperti *blueprint* yang membantu dalam pengolahan data pada bagian *back-end* server yang dihubungkan dengan PostgreSQL, yaitu *database* berorientasi objek.

3.3.2. Proses Pelaksanaan

Perancangan dilakukan menggunakan *database* PostgreSQL yang menggunakan Dalam perancangan *website* ini digunakan software dan hardware yang membantu penulis untuk melakukan tugasnya.

Software yang digunakan adalah:

1. Visual Studio Code versi 1.56.2
2. Sails.js versi 1.4.0
3. Ember.js versi 3.25.1

4. PostgreSQL versi 10.16
5. PouchDB versi 6.12.0
6. DBeaver versi 21.0.3.202104181339

Hardware yang digunakan adalah:

1. *Notebook* : Acer Aspire E 14.
2. Prosesor : Intel Core i5-7200U 2.5GHz.
3. Memori : 8GB RAM.

A. Flowchart

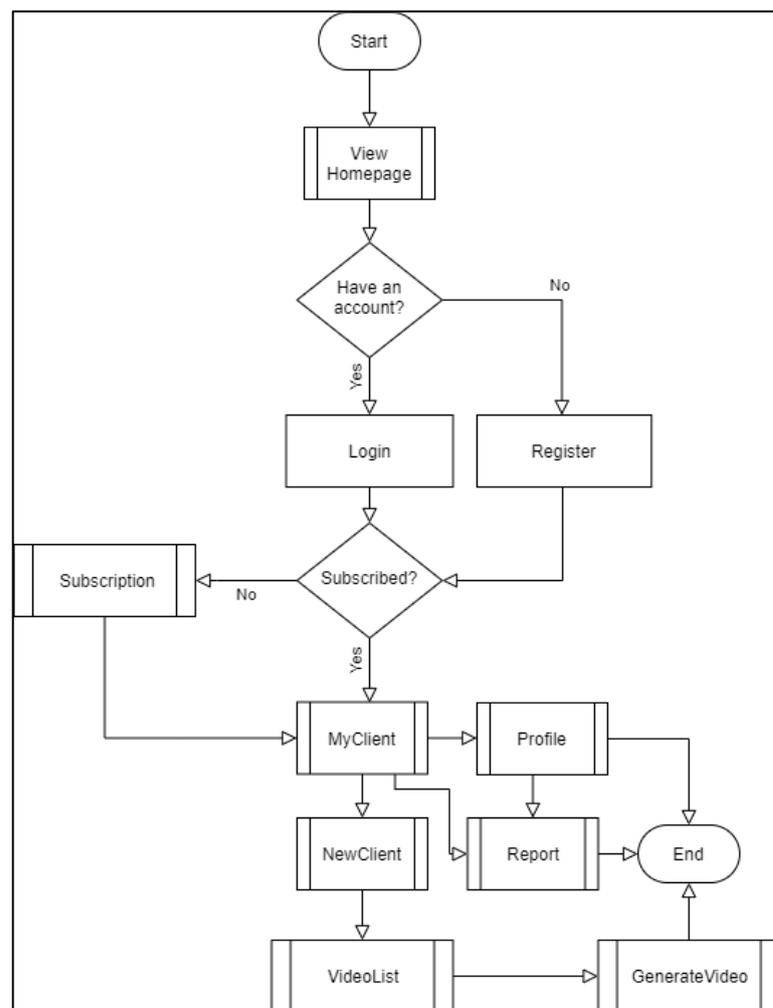
Cara kerja *website* ini dapat dijelaskan melalui *flowchart* berikut.

A.1. Flowchart Website

Website yang dibuat dinamakan “FinViews” yang merupakan website pembuatan video personalisasi yang dikhususkan untuk agen asuransi kepada *client* pengguna asuransi, agen asuransi dapat berlangganan servis ini dengan memilih penawaran yang berbeda-beda dalam waktu bulanan atau tahunan dan memiliki keuntungan yang berbeda-beda pada setiap *tier*-nya.

Pada saat mengakses website ini pertama kali yang ada halaman *homepage*, setelah itu pengguna akan bisa melakukan *login* atau *register*, setelah melakukan proses tersebut, akan dialihkan ke halaman *Myclient*, namun jika tidak melakukan langganan akan tidak akan muncul halaman *Myclient*, pengguna akan dialihkan ke halaman *Subscription*, selain itu pengguna bisa mengedit Profil mereka sesuai yang diinginkan di halaman profil dan jika ingin melihat laporan hasil interaksi agen dan *client* dapat membuka halaman profile.

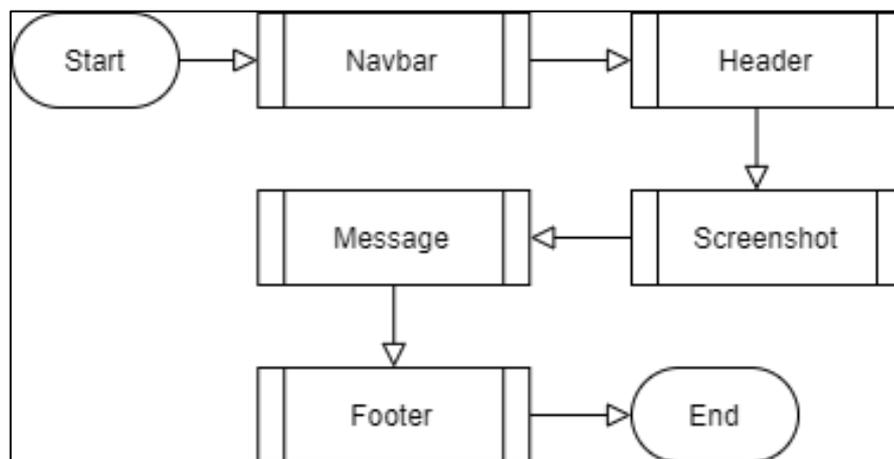
Halaman *Subscription* adalah halaman dimana pengguna bisa memilih kartu dan paket berlangganan yang sesuai dengan mereka, setelah memiliki langganan tampilan *Myclient* akan muncul dan pengguna bisa langsung melakukan personalisasi video dengan melakukan proses *NewClient* dimana pengguna akan mengisi informasi mengenai *client* mereka, setelah selesai mereka dapat memilih video yang diinginkan untuk dibuatkan *personalised video* dalam tampilan *VideoList*, setelah selesai memilih video akan diproses di halaman *GenerateVideo* dan video siap untuk dikirim kepada *client* yang bersangkutan. Flowchart website yang dibuat dapat dilihat dibawah pada gambar 3.1.



Gambar 3.1 Flowchart dari *website*

A.1.1. Flowchart view Homepage

Pada halaman Homepage, terdiri dari bagian *navbar* yang berisi *link* untuk *Pricing*, *Login* dan *Signup*, *header* yang hanya sekedar tampilan penyambut *user*, *screenshot* tempat *preview* fitur yang dimiliki “FinViews”, *message* yang digunakan untuk *feedback*, dan *footer* sebagai tempat menaruh informasi website; seperti yang ditunjukkan gambar 3.2.

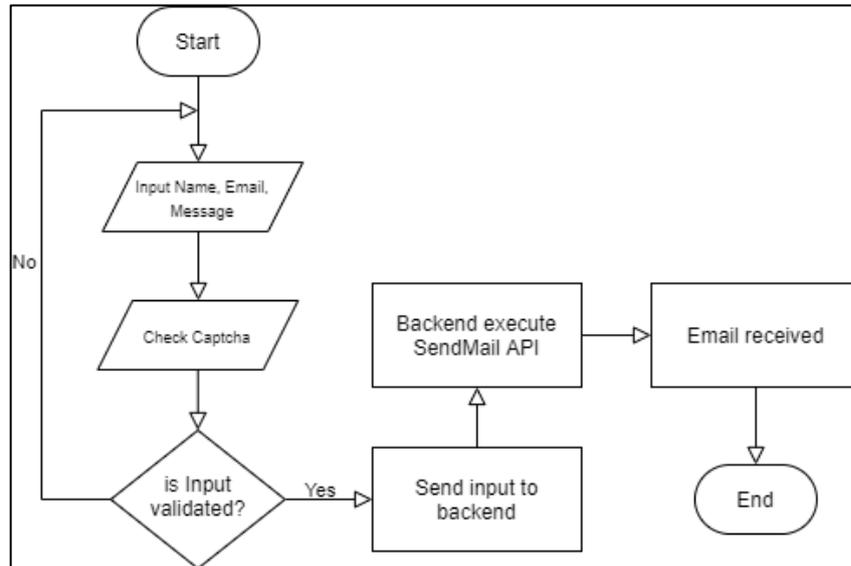


Gambar 3.2 Flowchart dari *view* Homepage

A.1.1.1. Flowchart view Message

Fitur *message* yang juga disebut *Message Us* ini menyediakan kirim pesan dari pengunjung baik itu saran atau kritik ataupun pertanyaan. Pengunjung harus mengisi nama, *email* yang valid dan pesan yang ingin disampaikan, setelah menekan *submit*, akan mengecek *captcha* jika berhasil maka data yang diisi akan dikirimkan ke bagian *backend server* dimana, di bagian *server* akan menjalankan API untuk mengirimkan *email* dari *email server* ke email perusahaan yang nantinya akan

digunakan untuk membalas pertanyaan yang telah diisi tadi melalui *email*. Berikut adalah Flowchart Message pada gambar 3.3.



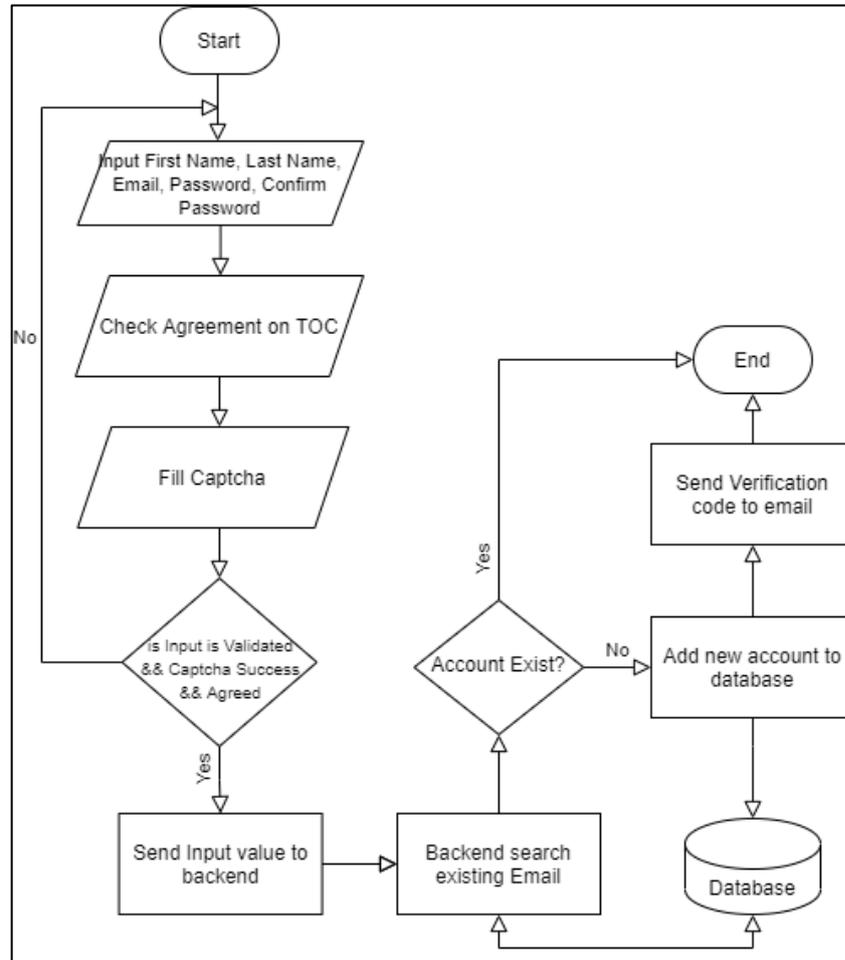
Gambar 3.3 Flowchart dari *view Message*.

A.1.2. Flowchart Register

Fitur Register terdapat pada *View Navbar* yang dapat diakses dimanapun, pada saat pengguna ingin mendaftarkan diri untuk menggunakan fitur-fitur dalam *website* ini, pengguna bisa menekan tombol “New User – Register” pada *Navbar*, pengguna diharuskan mengisi nama depan, nama belakang, alamat *email*, kata sandi dan konfirmasi kata sandinya, lalu juga harus menyetujui syarat dan kondisi dari *website* ini, setelah itu mengisi *captcha*, jika semua input tervalidasi maka data yang telah diisi akan dikirimkan ke bagian *backend server*.

Dalam bagian ini server akan mengecek keberadaan akun dari email ke *database website* ini, jika data ada maka akan mengulang pengisian data kembali,

jika berhasil akan mengirimkan kode verifikasi ke alamat *email* yang bersangkutan.
Berikut adalah Flowchart dari proses Register.

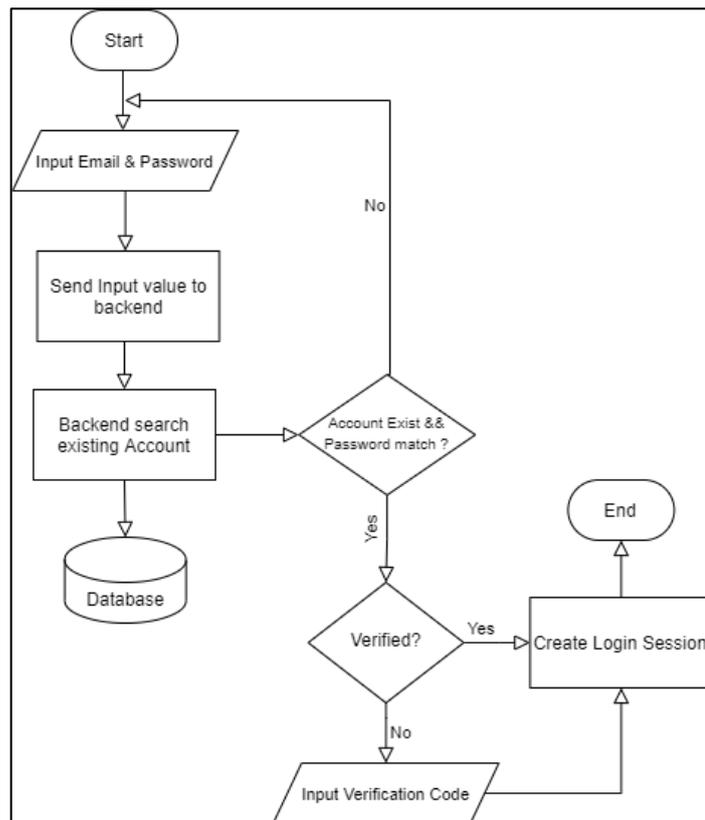


Gambar 3.4 Flowchart dari Register

A.1.3. Flowchart Login

Fitur Login juga terdapat pada *View* Navbar yang dapat diakses dimanapun, Jika pengguna sudah memiliki akun, pengguna bisa langsung melakukan *login*, dengan klik tombol “Login” pada Navbar, pengguna harus mengisi alamat *email* dan kata sandi mereka, setelah itu data tersebut akan dikirim ke bagian *backend server*. Server akan mengecek jika akun tersebut ada kepada *database* dan

mencocokkan kata sandi yang telah diisi dengan kata sandi yang terdapat pada *database* jika sudah akan mengecek status akun yang sudah mengisi kode verifikasi jika belum pengguna harus mengisi kode verifikasi yang telah diterima di alamat email, jika sudah server akan membuat sesi *login* dari pengguna, seperti yang pada Gambar 3.5.

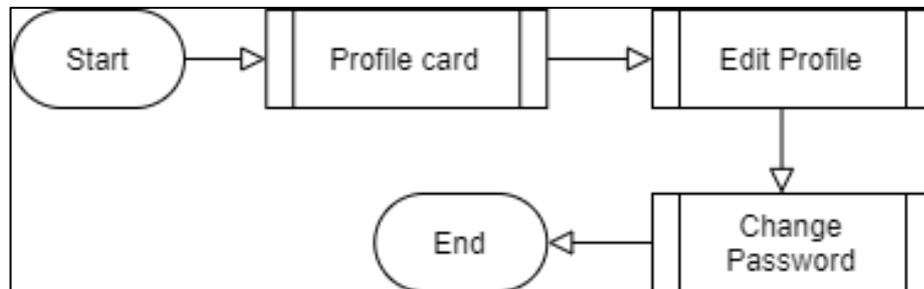


Gambar 3.5 Flowchart Login

A.1.4. Flowchart view Profile

Pada halaman ini cukup *simple* dihalaman ini terdapat Profile *card* yang menunjukkan informasi pengguna setelah masuk seperti nama dan *icon* lalu ada, *edit profile* yang dapat melakukan perubahan informasi profil *user* ditempat secara

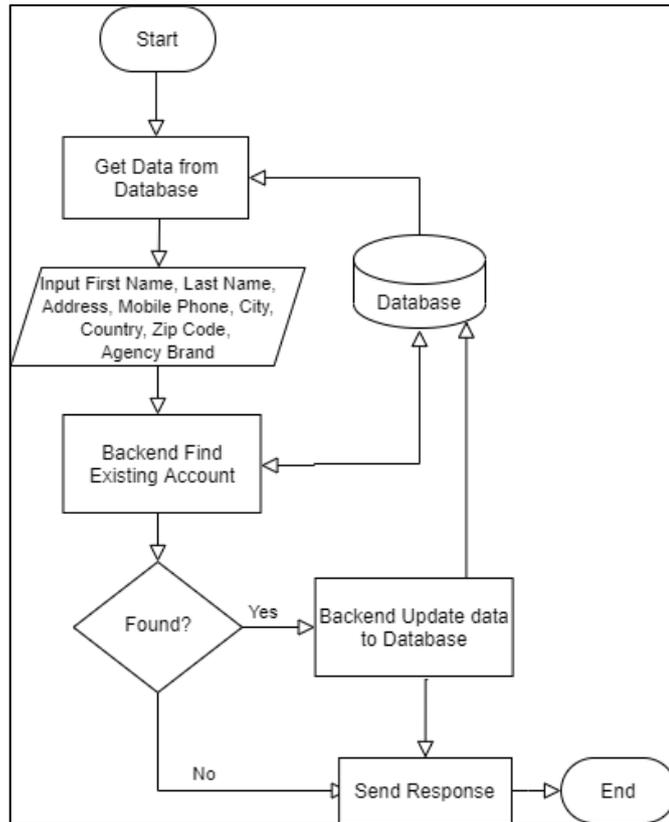
cepat, namun perlu diketahui untuk *email* tidak bisa diganti dan *change password* dimana *user* bisa mengganti kata sandi dengan mengisi kata sandi baru dan konfirmasi kata sandi baru seperti yang ditunjuk pada Gambar 3.6.



Gambar 3.6 Flowchart dari *view* Profile

A.1.4.1. Flowchart view Edit Profile

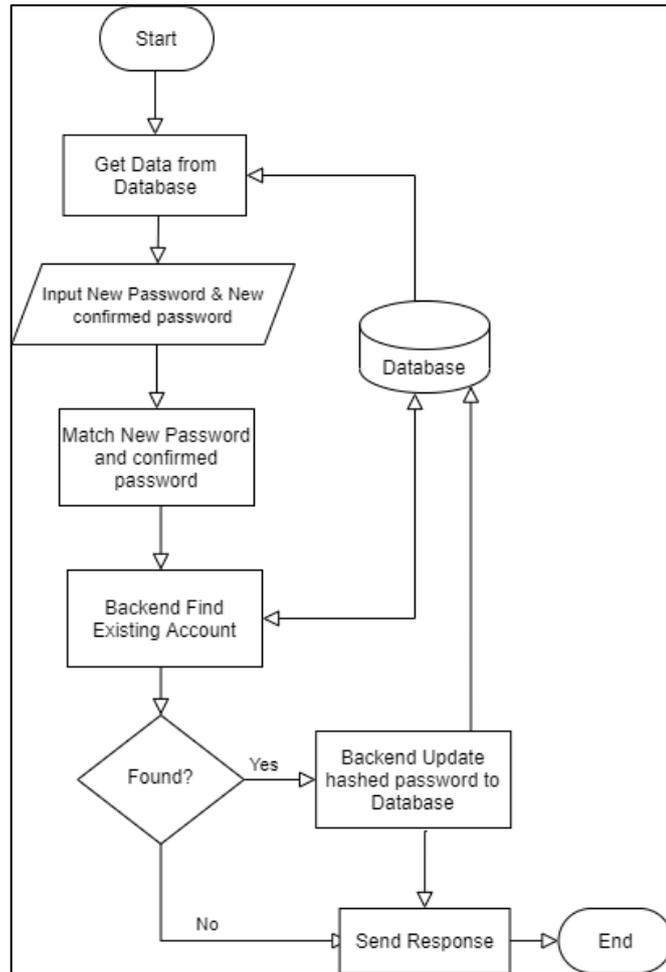
Pada saat pengguna masuk ke halaman ini *website* akan meminta data dari *database* dengan API dari *backend server* setelah itu, data akan muncul di dalam *input*, pengguna bisa mengisi kolom input kecuali alamat *email*, setelah menekan tombol *submit* data akan dikirim ke *backend server* dan server akan mencari akun yang ada dari *database* jika ditemukan maka server akan melakukan *update* pada akun yang telah ada sebelumnya dan mengirim respons ke bagian *frontend*, namun jika akun tidak ditemukan maka akan memberikan respons *error* ke *frontend*, yang dapat dilihat pada Gambar 3.7.



Gambar 3.7 Flowchart Edit Profile

A.1.4.2. Flowchart view Edit Password

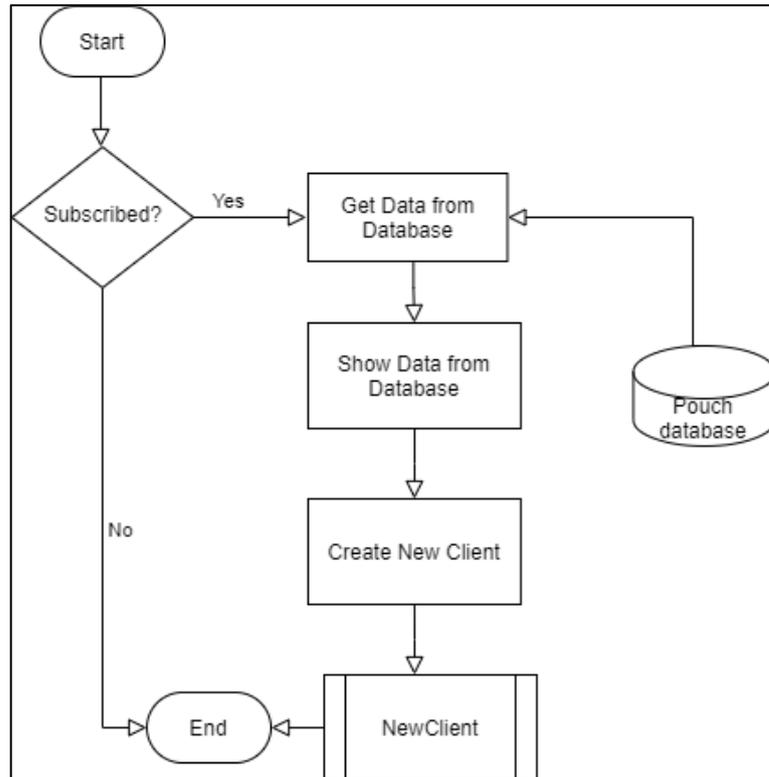
Pada fitur *change password* pengguna mengisi kata sandi baru yang dinantinya akan dikonfirmasi dengan mengisi kata sandi yang sama pada kolom *re-type password* setelah itu kata sandi akan dicocokkan dan dikirim ke *backend server*, di server akan mencari akun yang eksis pada *database* dan jika ditemukan maka akan dihash dan disimpan di *update* pada *database* lalu mengirim respons ke *frontend* jika gagal akan mengirim respons *error* ke *frontend*, berikut gambar Flowchart Edit Password.



Gambar 3.8 Flowchart Edit Password

A.1.5. Flowchart My Clients

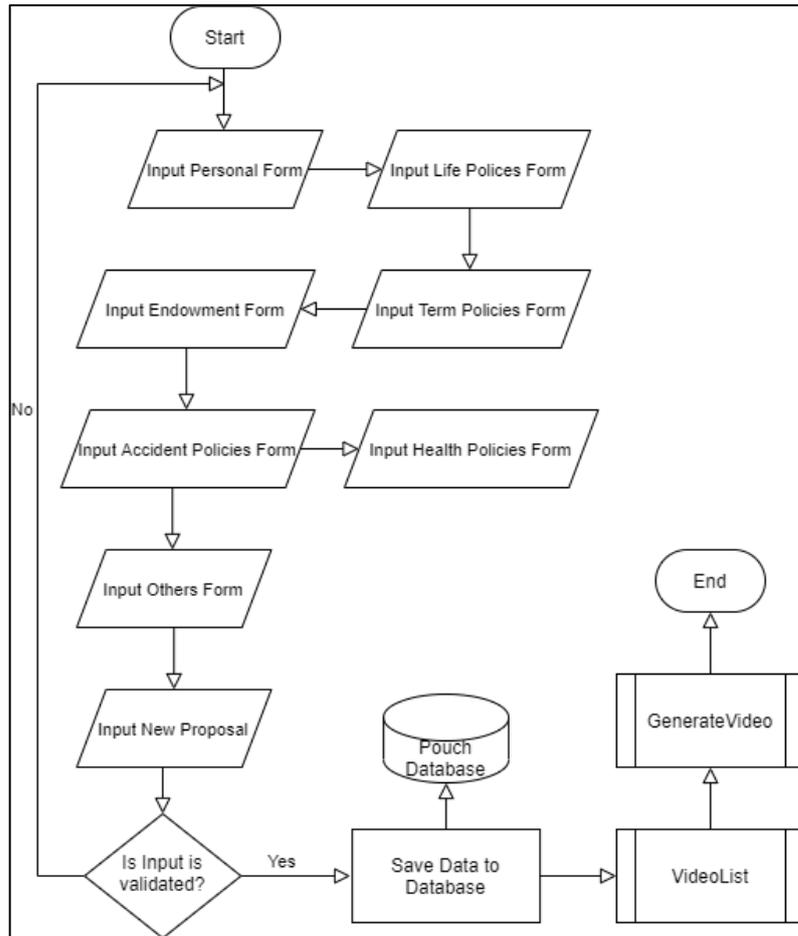
Pada saat halaman My Clients dibuka jika pengguna tidak melakukan *subscription*, halaman akan tertutup dengan pemberitahuan untuk melakukan *subscription* jika sudah melakukan *subscription*, maka halaman akan muncul. Halaman akan meminta data dari Pouch DB dan menampilkannya dan dari sini pengguna bisa menambahkan *client* baru, seperti Gambar 3.9.



Gambar 3.9 Flowchart My Clients

A.1.6. Flowchart New Client

Halaman *NewClient* memuat formulir-formulir yang harus diisi sesuai dengan *client* yang bersangkutan, seperti data diri dan kebijakan bersama pihak asuransi terkait, jika data sudah selesai diisi dan divalidasi maka data akan disimpan ke PouchDB, pengguna akan dialihkan ke halaman pemilihan video setelah memilih video terkait akan dibawa ke halaman “generate video” dimana pengguna bisa melihat langsung *progress* dari pembuatan video tersebut, yang dapat dilihat pada gambar 3.10.



Gambar 3.10 Flowchart New Client

B. Struktur Tabel

Tabel yang digunakan website ini menggunakan 2 *database* yaitu PostgreSQL (gambar 3.2) yang merupakan *database* berbasis objek untuk menyimpan data agen yang sudah mendaftar dan PouchDB (gambar 3.3) yang merupakan *database NoSQL* lokal, untuk menyimpan data klien, karena FinViews merupakan pihak ketiga, server tidak menyimpan data klien jadi data klien berada pada lokal *database* dari setiap peramban yang digunakan oleh agen.

Tabel 3.2 Struktur Tabel *Agents*

Nama Kolom	Tipe Data
id (PK)	Serial
fname	varchar
lname	varchar
email	varchar
password	varchar
last_outreach	timestampz
dob	timestamp
total_video	int
createdat	timestampz
updatedat	timestampz
verificationcode	text
totalclient	int
totalearning	int
totalspent	int
agentnumber	varchar
address	varchar
city	varchar
country	varchar
zipcode	varchar
agency_brand	varchar
mobile_number	varchar

Untuk tabel *clients* dapat dilihat melalui tabel 3.3 pada halaman lampiran.

B.1. Entity Relationship Diagram

Relasi antara database *Agents* dan *Clients* adalah *one to many* dimana agen bisa memiliki banyak klien, namun klien hanya bisa mendaftar pada satu agen asuransi saja, misal satu klien hanya boleh memiliki satu agen asuransi perorangan yang didaftarkan pada *FinViews* ini pada gambar 3.11.



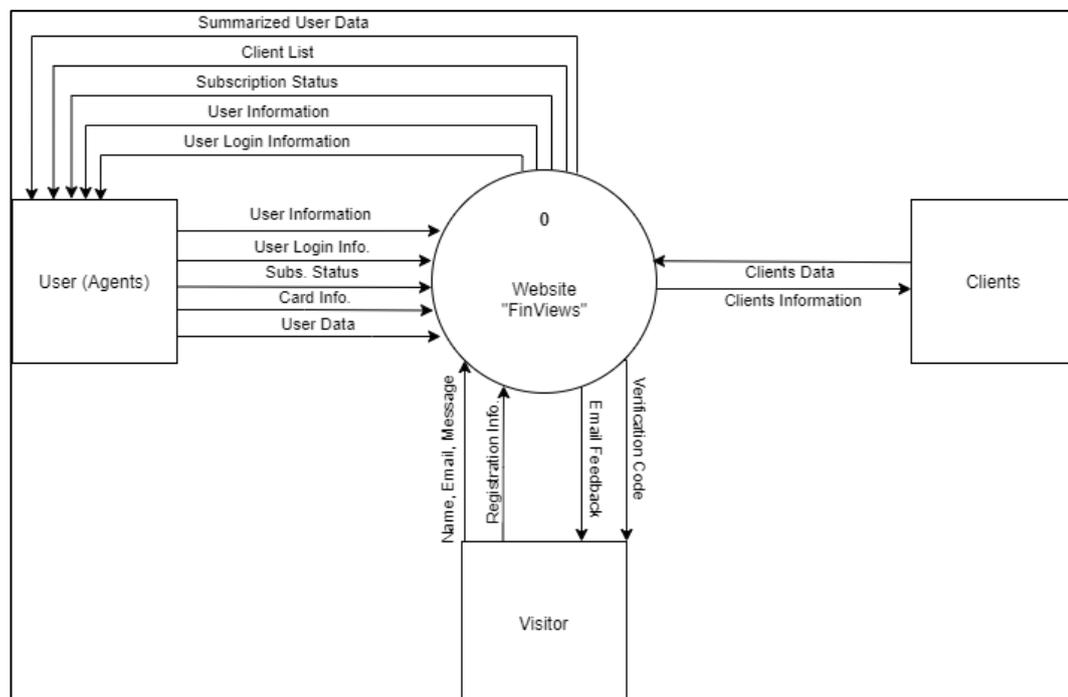
Gambar 3.11 Relasi antara *Database* Agents dan Clients

C. Data Flow Diagram

Data flow diagram merupakan peta aliran data yang berpindah-pindah dari suatu sistem ke sistem yang lain, dalam laporan ini digunakan notasi Yourdan dan Demarco.

C.1. Context Diagram

Website ini memiliki 3 *external entity*, *Visitor* sebagai pengunjung yang tidak memiliki akun yang hanya bisa melakukan kirim pesan, namun bisa register untuk menjadi pengguna; *User* yang disebut agen (asuransi) yang mendaftar pada aplikasi ini dan bisa mengisi data klien yang mereka miliki, melakukan pembayaran untuk *subscription* dan mengedit profil mereka; *Clients* yang memberikan datanya.



Gambar 3.12 Context Diagram

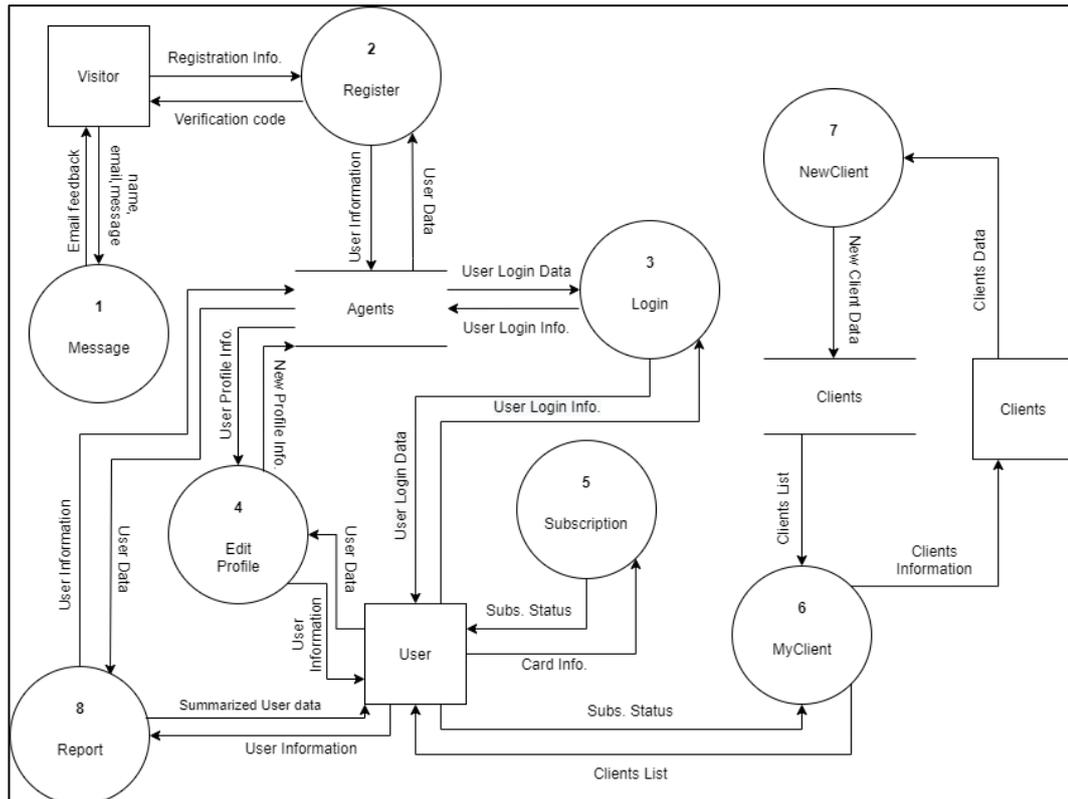
C.2. Data Flow Diagram Level 0

Dalam website ini pengunjung yang mengunjungi website ini bisa mengirim data mereka yang berupa nama, alamat *email*, dan pesan mereka ke Proses Message sementara dari proses ini akan diberikan lagi *feedback* dari pesan yang dikirim oleh pengunjung. Pengunjung bisa memberikan informasi mereka ke Proses Registrasi dan bisa mendapatkan kode verifikasi, dari Proses Registrasi, informasi yang telah diberikan akan dimasukkan ke dalam *database* Agents, dari sini pengguna dapat menerima data yang mereka daftarkan.

Pengguna bisa melakukan Proses Login dengan mengirim informasi *login* yang dibutuhkan, proses ini akan mengembalikan data *user* kepada pengguna dari *database*. Jika ingin melakukan perubahan pada akun, pengguna dapat mengirim informasi baru ke Proses Edit Profil yang nantinya akan disimpan dalam *database*, dari proses ini pengguna juga dapat menerima data pengguna yang terkini. Untuk Proses Subscription pengguna dapat mengirim informasi mengenai kartu metode pembayaran yang akan digunakan untuk membayar paket langganan sementara itu Proses Subscription akan mengembalikan status berlangganan dari pengguna.

Untuk Proses *MyClients* pengguna akan menerima daftar klien dari *Database Client*, dan dari proses ini mengirimkan informasi pengguna untuk mengolah mana yang menjadi klien dari pengguna tersebut. Proses *NewClient* akan membuat klien baru yang dipasangkan dengan pengguna dengan mengirimkan informasi klien yang nantinya akan disimpan ke dalam *database Client*.

Proses *Report* akan mengambil data dari *database* Agent dan menampilkan nya ke agen itu sendiri sebagai rangkuman informasi yang mereka miliki, sebagaimana yang dapat dilihat melalui Gambar 3.12 Data Flow Diagram.



Gambar 3.13 Data Flow Diagram level 0

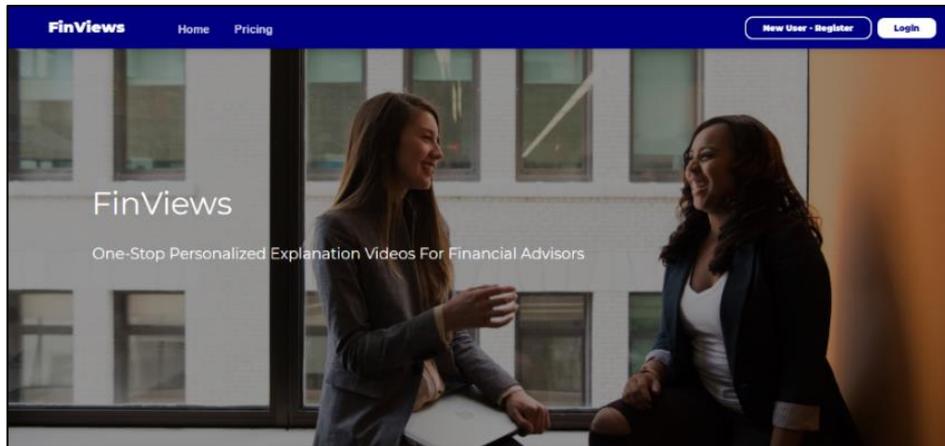
D. Implementasi

Berikut merupakan hasil dari implementasi dan beberapa tampilan *website*.

D.1. Homepage

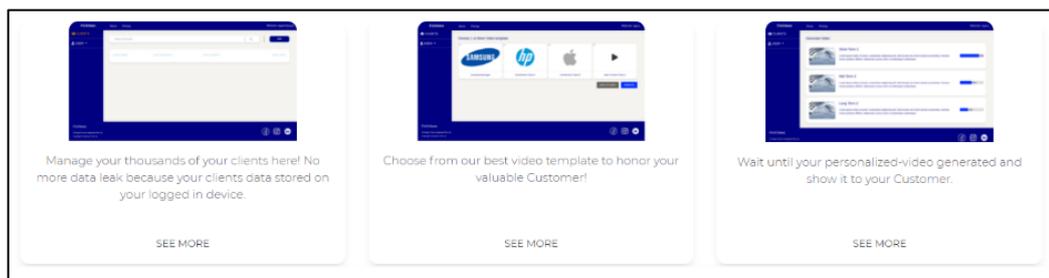
Halaman Homepage dibagi menjadi beberapa *section* yang pertama adalah *navbar* dimana terdapat judul *website* dengan sub-judul “Home” yang mengarah ke

halaman utama dan “Pricing” yang mengarah ke halaman harga dan tombol untuk *login* dan *register*, setelah itu ada *header* dimana ada nama aplikasi dan slogan yang dimiliki dengan *background* gambar, dapat dilihat pada gambar 3.14.



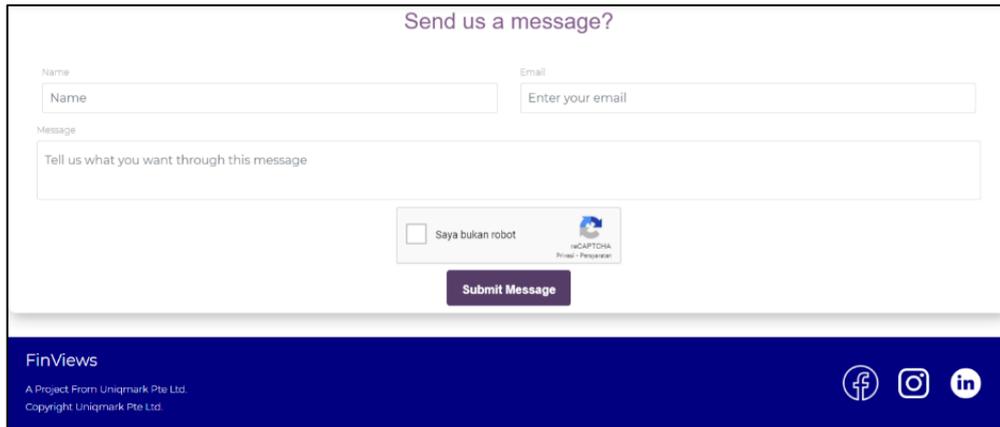
Gambar 3.14 Tampilan Header pada Homepage.

Setelah itu dilanjutkan dengan cuplikan gambar dari fitur-fitur yang ada dalam website ini, yang dapat dilihat pada Gambar 3.15 terdapat gambar halaman *MyClients*, *VideoList*, dan proses *Generate Video*.



Gambar 3.15 Tampilan screenshot fitur aplikasi.

Setelah itu ada tampilan “Send us a message?” dimana pengguna bisa mengirimkan pesan, kritik, saran dan pertanyaan kepada website owner yang nantinya akan dibalas kembali melalui surel dengan mengisi nama, *email* yang bisa dihubungi dan pesannya serta mengisi *captcha* terlebih dahulu.



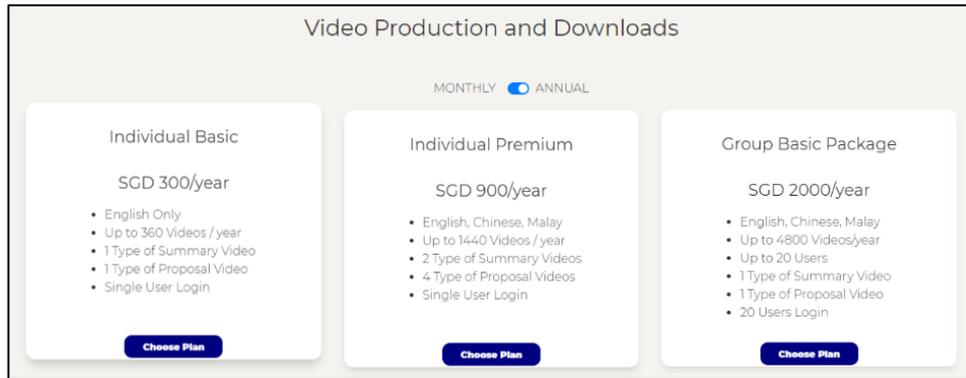
Gambar 3.16 Tampilan “Send us a message?”

D.2. Pricing

Pricing merupakan halaman pada website ini yang memberikan informasi mengenai tingkatan dan biaya dari langganan servis pada FinViews, untuk waktunya ada dua, yaitu bulanan dan tahunan yang dapat diganti melalui tombol *toggle*, untuk tipe langganan ada tiga tipe, ada “Individual Basic” yang menawarkan sampai 30 video dalam bahasa inggris saja; ada “Individual Premium” yang menawarkan 30 video dengan tiga bahasa yang dapat digunakan; ada “Group Basic Package” yang menawarkan 200 video dengan bahasa inggris yang dapat digunakan lebih dari satu akun, sama juga dengan yang tahunan.



Gambar 3.17 Tampilan harga bulanan pada Pricing



Gambar 3.18 Tampilan harga tahunan pada Pricing

D.3. Register

Jika pengguna tidak memiliki akun, pengguna tidak bisa mengakses fitur yang berada pada website ini, pengguna bisa melakukan registrasi dengan menekan tombol “New User – Register” pada *navbar*, setelah itu akan muncul modal *Register* dimana pengguna bisa mengisi informasi yang dibutuhkan untuk membuat akun pada website ini.

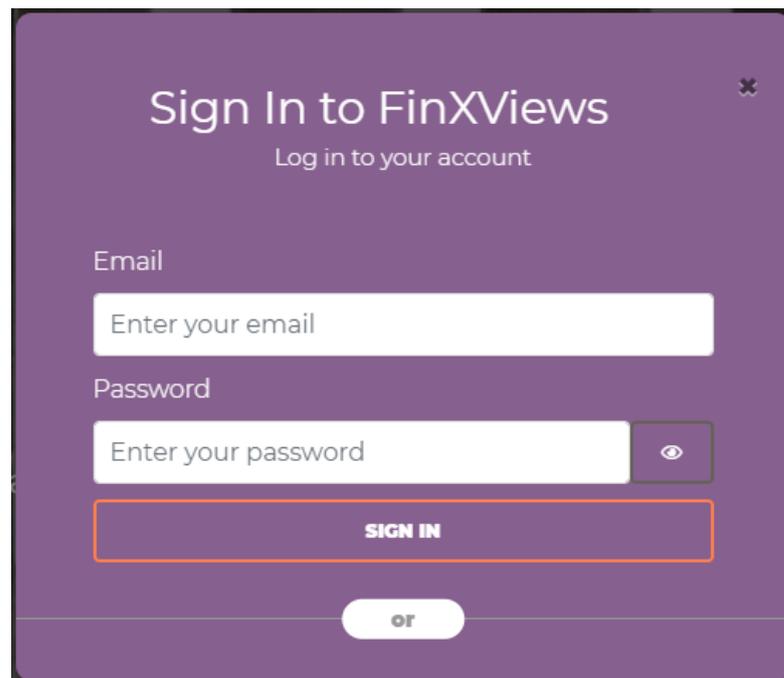
The image shows a registration form titled 'Sign Up to FinXViews'. The form includes the following fields and elements:

- First Name:
- Last Name:
- Email:
- Password:
- Confirm Password:
- Agreement: I have read and agree with the [Terms and condition](#) from FinXViews
- reCAPTCHA: Saya bukan robot
- Buttons: and [Already registered? Click here to sign in](#)

Gambar 3.19 Tampilan modal Register

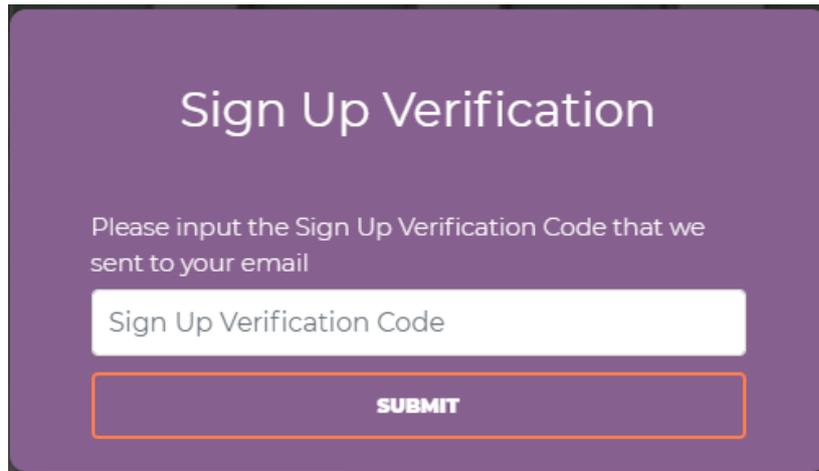
D.4. Login

Jika sudah memiliki akun pada website ini, pengguna bisa masuk dengan menekan tombol “Login” pada *navbar* setelah itu akan muncul modal tampilan *login*, pengguna bisa mengisi alamat *email* yang sudah terdaftar dan mengisi kata sandi, yang dapat dilihat pada gambar 3.20.



Gambar 3.20 Tampilan modal Login

Jika pengguna belum melakukan verifikasi kode akan muncul *modal* baru yang mengharuskan untuk pengguna melakukan verifikasi, *email* yang berisi kode akan diberikan ke *email user* yang digunakan untuk mendaftar di website ini, kode ini berlaku selama 30 menit, instruksi juga akan diberikan bersamaan dengan kode yang dikirim

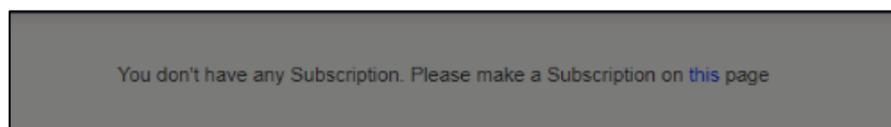


Gambar 3.21 Tampilan modal verifikasi Login

Jika sudah melakukan verifikasi pengguna akan langsung dibawa ke halaman “MyClients”

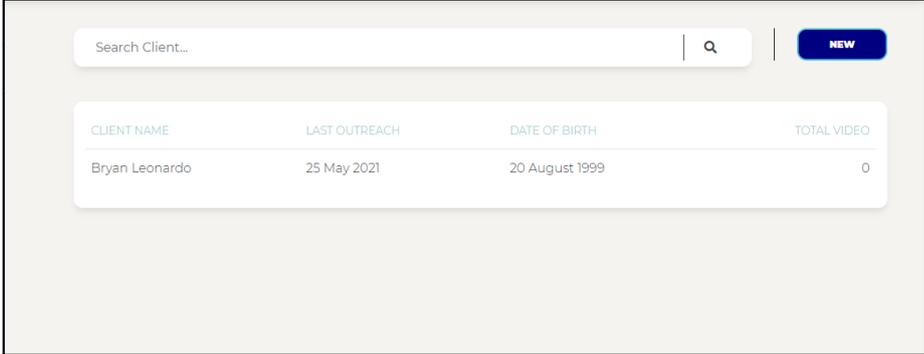
D.5. My Clients

Pada awal halaman jika pengguna belum melakukan langganan untuk fitur akan muncul teks “You don’t have any Subscription. Please make a Subscription on this page” yang mencegah *user* untuk melihat dan menambahkan data ke *MyClients*, pengguna bisa klik teks “this” yang akan mengantar ke halaman *Subscription* seperti Gambar 3.22



Gambar 3.22 Tampilan pengguna non-langganan

Jika sudah berlangganan, akan muncul table yang nantinya berisi *client* yang telah didaftarkan oleh agen, ada juga *search bar* yang akan digunakan untuk mencari klien.



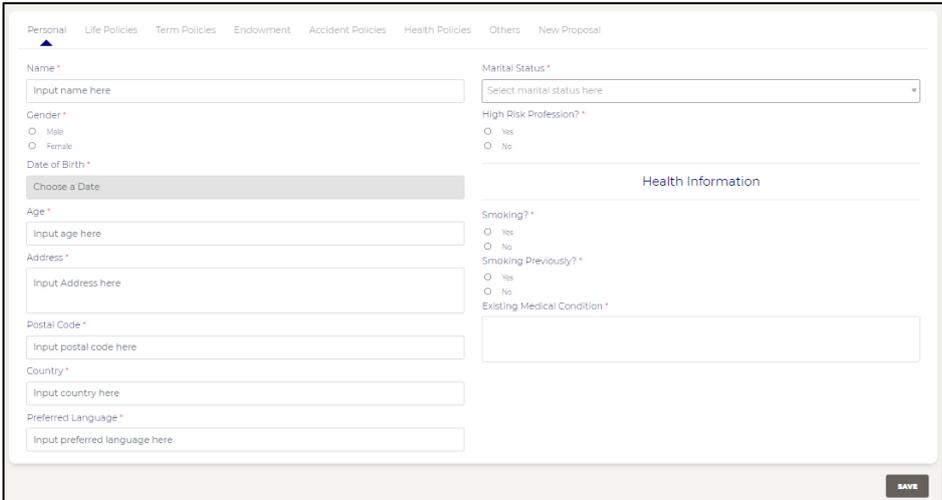
The screenshot shows a web interface for managing clients. At the top, there is a search bar labeled "Search Client..." with a magnifying glass icon and a blue "NEW" button. Below the search bar is a table with the following data:

CLIENT NAME	LAST OUTREACH	DATE OF BIRTH	TOTAL VIDEO
Bryan Leonardo	25 May 2021	20 August 1999	0

Gambar 3.23 Tampilan MyClients

D.6. New Clients

Pada halaman ini terdapat banyak form yang harus diisi pengguna mengenai klien-kliennya yang bersangkutan, setelah itu bisa *submit* dan diolah untuk menjadi *personalized video*.



The screenshot shows the "Personal" section of a form for creating a new client. The form is divided into two columns. The left column contains fields for Name, Gender (radio buttons for Male and Female), Date of Birth (with a "Choose a Date" button), Age, Address, Postal Code, Country, and Preferred Language. The right column contains fields for Marital Status (dropdown), High Risk Profession? (radio buttons for Yes and No), and Existing Medical Condition (text area). A "SAVE" button is located at the bottom right of the form.

Gambar 3.24 Tampilan form bagian Personal pada NewClient

Gambar 3.25 Tampilan “Life Policies” form pada halaman “NewClients”

Untuk form “Life Policies”, “Term Policies”, “Endowment”, “Accident Policies”, “Health Policies” menggunakan formulir dinamis, dimana pengguna menuliskan beberapa form yang dibutuhkan dan form itu akan muncul sesuai yang diisi pada kolom nomor *policy* (gambar 3.26)

Gambar 3.26 Tampilan saat sudah mengisi jumlah form

Panel-panel tersebut dapat diklik dan akan membuka form yang harus diisi (Gambar 3.27)

Gambar 3.27 Tampilan saat membuka *accordion* form

Jika sudah terisi dengan betul data akan disimpan ke dalam PouchDB.

D.7. Profile

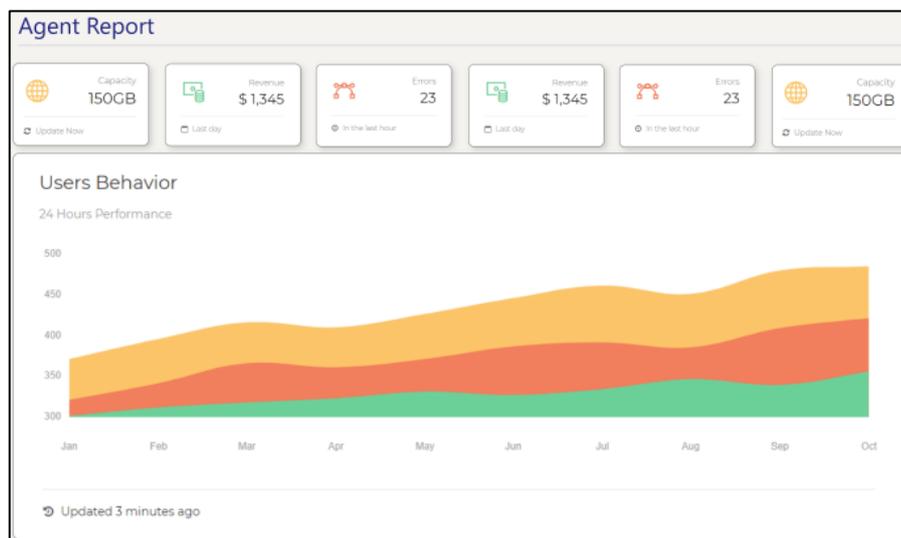
Halaman ini dapat digunakan pengguna untuk mengedit dan memperbaharui akun *user*, bagian profil dibagi menjadi per-kolom dan baris, baris yang pertama adalah *kartu* profile dari agen yang menunjukkan nama dan foto profil dan baris kedua, terdapat edit profil pada kolom pertama dan terdapat change password pada kolom kedua.

Gambar 3.28 Tampilan Profile

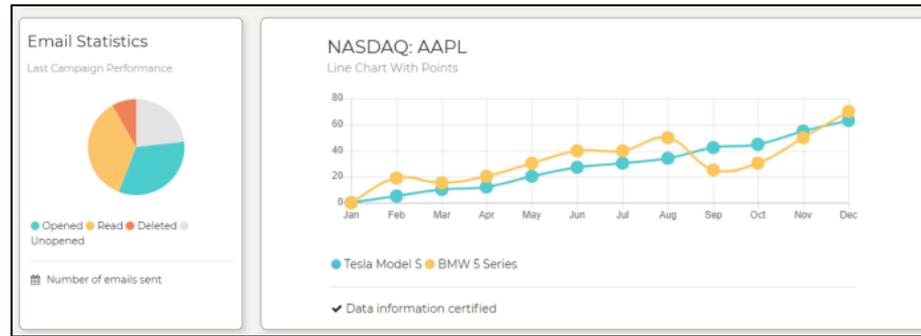
Gambar 3.29 Tampilan “Change Password” pada Profile

D.8. Report

Di halaman ini pengguna bisa melihat hasil rangkuman akun dari pengguna (agen) dimulai dari pemakaian kapasitas, keuntungan dan masalah-masalah, selain itu ada pengguna juga bisa melihat interaksi terhadap *clientnya* yang ditampilkan dalam bentuk grafik, seperti *user behavior*, *email* yang telah dibaca oleh *client* serta tingkatan saham dalam pembangunan dari grafik ini dibantu oleh *add-on Chart.js*.



Gambar 3.30 Tampilan “Report”

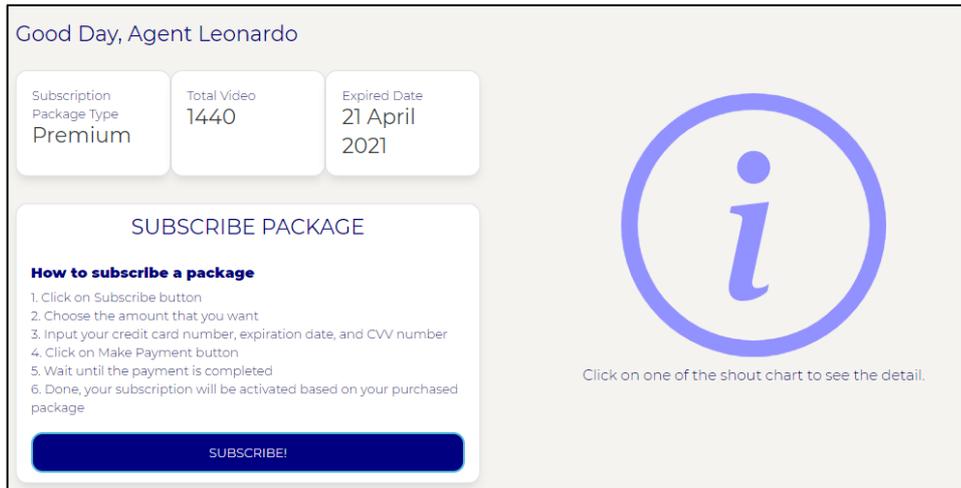


Gambar 3.31 Tampilan grafik pada “Report”

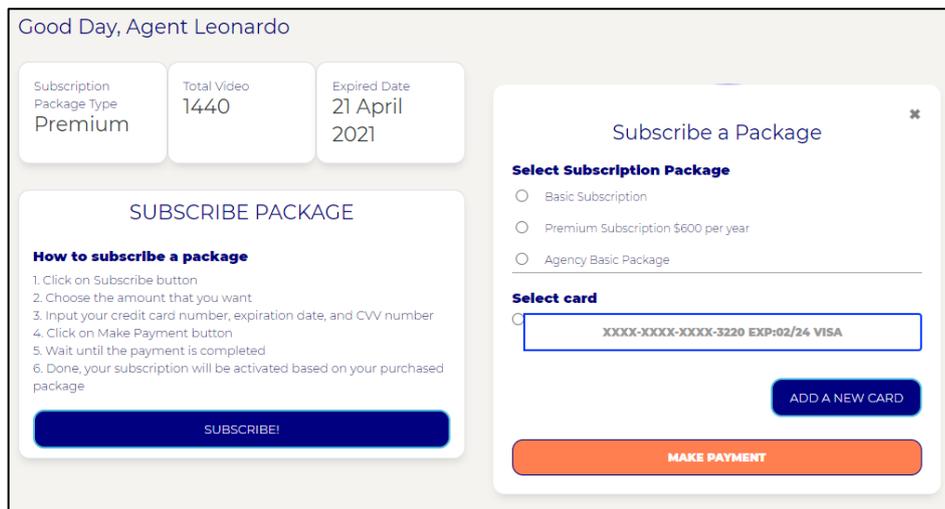
D.9. Subscription

Halaman ini akan pengguna bisa melakukan pembayaran untuk berlangganan menggunakan servis ini, dimulai dengan (gambar 3.32) mengeklik tombol “Subscribe!” setelah itu pengguna bisa melakukan berlangganan dengan memilih paket mulai dari “Basic Subscription”, “Premium Subscription”, yang terbagi menjadi bulanan dan tahunan dan dan “Agency Basic Package” untuk banyak orang (gambar 3.33).

Pengguna bisa memilih kartu debit yang akan digunakan untuk membayar atau menambahkan kartu baru dengan menekan tombol “Add A New Card” setelah itu akan muncul *modal* (gambar 3.34) dimana pengguna bisa mengisi informasi kartu mereka, jika sudah terisi akan muncul kartu baru. Jika ingin melanjutkan klik tombol “Make Payment” dan akan muncul *modal* konfirmasi pembayaran (gambar 3.35), setelah terkonfirmasi akan muncul modal sukses (gambar 3.36)



Gambar 3.32 Tampilan “Subscription”



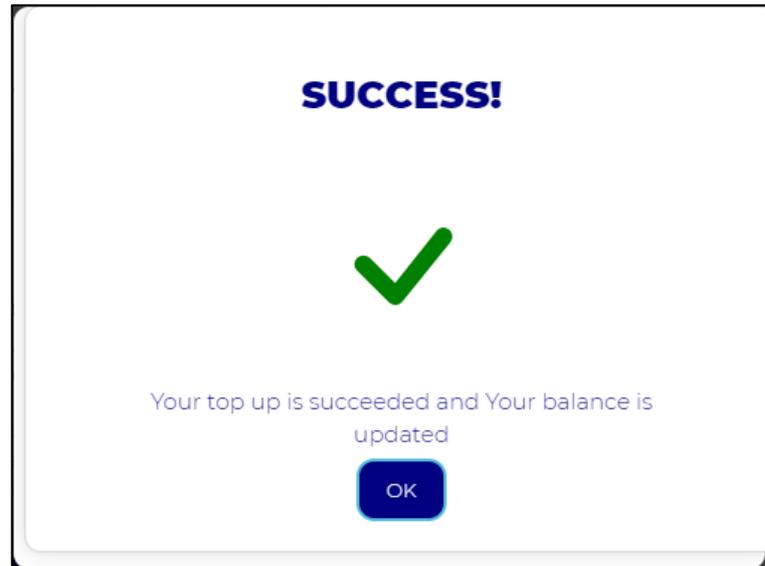
Gambar 3.33 Tampilan melakukan *subscription*

The image shows a modal window titled "ADD CARD" with a close button (X) in the top right corner. The form contains three input fields: a large field for "Card Number", and two smaller fields for "Expired Date" and "CVV". Below the input fields is a prominent blue button labeled "ADD CARD".

Gambar 3.34 Tampilan modal “Add Card”

The image shows a modal window titled "Payment Confirmation". The text inside reads: "You are about to make a transaction with this details below.", "Amount: XX", and "Payment method: Credit Card". Below this information is a warning: "Please make sure you are using correct card to this transaction". At the bottom, there are two buttons: a blue "CONFIRM" button and a white "CANCEL" button with a blue border.

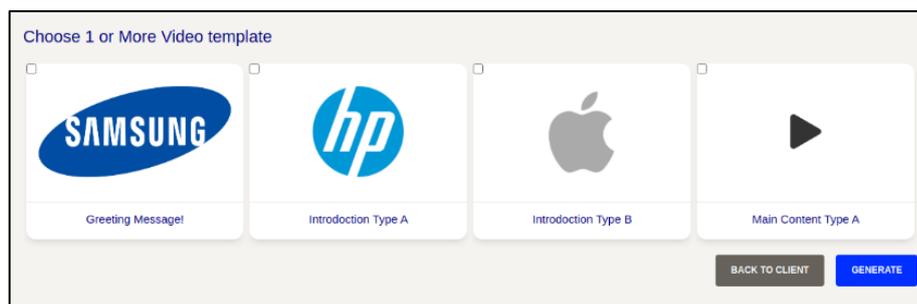
Gambar 3.35 Tampilan modal konfirmasi pembayaran



Gambar 3.36 Tampilan modal sukses

B.9. Video List

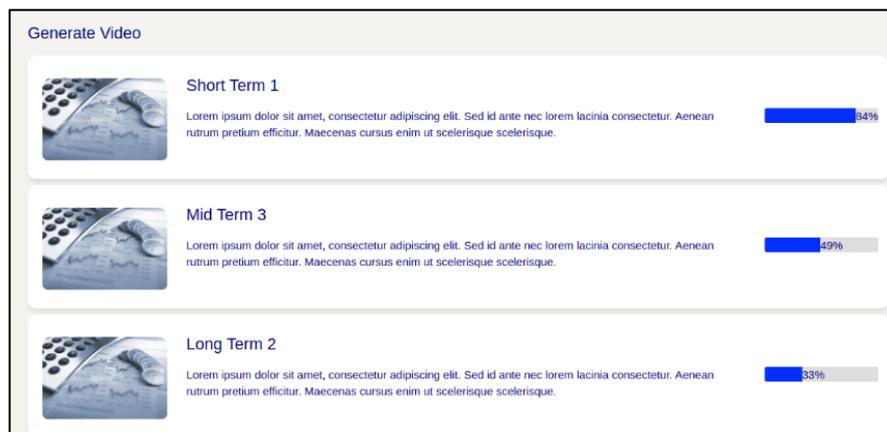
Halaman ini akan menunjukkan video-video yang tersedia yang bisa dipilih oleh pengguna untuk dijadikan *personalized* video, dalam praktek magang ini saya hanya membuat bagian *front-end*, untuk videonya akan dilanjutkan menggunakan Adobe Animation yang dikerjakan oleh departemen lain.



Gambar 3.37 Tampilan “Video List”

B.10. Generate Video

Halaman ini akan menunjukkan proses pembuatan video personalisasi yang sedang dikerjakan, berisi *cards* yang didalamnya ada gambar video yang dipilih, judul video yang dipilih serta deskripsinya dan juga ada *progress bar* dengan persenan untuk menunjukkan sampai mana proses video itu dibuat.



Gambar 3.38 Tampilan “Generate Video”

3.3.3. Kendala yang ditemukan

Dalam rancang bangun *website* ini ditemukan beberapa kendala yang menghalangi dalam dikembangkannya *website* ini seperti berikut.

- Cepatnya perkembangan *Ember.js* dan *Sails.js*. membuat sulit untuk mengikutinya, ditambah dengan adanya *Ember Octane* dimana beberapa kode ada yang berbuah secara drastis dan ada yang dibuang.
- Perubahan rencana ditengah-tengah *development*.
- Perbedaan *environment* yang digunakan, selama melakukan magang penulis menggunakan *localhost* pribadi yang nantinya baru akan digabungkan dengan *server*.

3.3.4. Solusi atas kendala yang ditemukan

Dari kendala-kendala tersebut, penulis dapat menemukan solusi yang membantu memecahkan masalahnya, yaitu:

- a. Mencari informasi di internet dan bertanya mengenai perbedaan *Ember Octane* dan *Ember Classic* serta informasi mengenai *Sails.js* terbaru.
- b. Melakukan perubahan sesuai yang di beritahukan secepatnya.
- c. Setelah melakukan *coding*, hasil pekerjaan diberikan kepada pembimbing untuk digabungkan dengan server utama.