Langkah pertama, yang dibuat adalah kita membuat server dari suatu website Insframe tersebut bernama sever.js dan kita mendownload node modules terlebih dahulu dan kita menginstall npm yang diperlukan yaitu :

* "bcrypt": "^5.0.0",
* "body-parser": "^1.19.0",
* "cloudinary": "^1.23.0",
* "cookie-parser": "^1.4.5",
* "cors": "^2.8.5",
* "dotenv": "^8.2.0",
* "e": "^0.1.3",
* "ejs": "^3.1.5",
* "express": "^4.17.1",
* "express-ejs-layouts": "^2.5.0",
* "express-session": "^1.17.1",
* "joi": "^17.3.0",
* "jsonwebtoken": "^8.5.1",
* "moment": "^2.29.1",
* "mongoose": "^5.10.14",
* "multer": "^1.4.2",
* "request": "^2.88.2"

Langkah kedua, yang kita buat route di InsframeContent.js untuk melakukan interaksi dalam website tersebut seperti menampilkan tampilan home kita dengan membuat url di atas dapat disampingkan dengan tampilan yang lain seperti mencari category, popular dan lain-lain. Kita membuat router (/popular) untuk menampilkan tampilan popular dari website tersebut dengan melakukan sorting dari views foto tersebut, (/category) untuk menampilkan pilihan category yang teresdia, (/leaderboard) untuk menampilkan author yang paling banyak memiliki viewer dalam fotonya, (/category/:categoryName) -> (categoryName merupakan nama category tersebut bukan id dari category. Mengambil dari database) mengambil data foto-foto yang berada di category tersebut, (/photo/:photoName) -> (photoName menggunakan id photo tersebut mengambil dari backend) mengambil data untuk di pop apabila gambar tersebut di klik dan menambah function apabila gambar tersebut diklik maka dia akan menambahkan view dari foto tersebut, (/upload) untuk melakukan upload tetapi user harus sudah log in kepada insframe tersebut.(/form-upload) untuk membuat deskripsi dalam foto yang akan diupload dan menentukan category untuk foto tersebut, (/confirm-upload) untuk mengambil database dari foto yang diupload dan (/search) untuk dapat melakukan pencarian foto baik dari keyword user, category dan title foto tersebut.

Langkah ketiga, membuat ejs dimulai dari partial-partial yang merupakan bagian dari website tersebut yang akan dipakai banyak oleh ejs yang lain seperti navbar , atau tampilan home yang memiliki sama dengan tampilan popular. Untuk navbar yang kita buat berada di layout.ejs dan banner-search.ejs tersebut untuk tampilan search yang beda karena tampilan search di home beda dengan yang lain. collage.ejs untuk menampilan foto. Tab.ejs untuk menampilkan tab kecil pilihan untuk pindah ke home, leaderboard, category dan popularity. Setelah partial dan layout telah selesai semua. Kita akan membuat ejs untuk pilihan category dengan select-category.ejs dengan mengambil partial tab.ejs dan collage.ejs dan kita menampilkan category yang kita buat di route tersebut untuk menampilkan category apa saja yang kita buat. Untuk popular tidak jauh beda dengan home karena hanya mengambil tab.ejs dan collage.ejs tetapi melakukan sorting dari view jadi tampilan urutan foto tersebut berbeda dengan home. 404.ejs untuk apabila suatu tampilan error dan pop-up.ejs mengambil database berupa deskripsi dari foto tersebut dan foto yang ingin dilihat, profile author dan lain-lain. Leaderboard.ejs mengambil tab.ejs lagi dan melakukan tampilan dari database dengan mengambil banyaknya views dan foto image profile author dan nama nya. Search.ejs mencari database yang dengan keyword query yang telah kita buat di route InsframeContent.js . login.ejs untuk melakukan login terhadap account insframe dan membuat sign up tersebut.

Langkah keempat, Membuat models yang berupa database yang kita buat di MonggoDB untuk category kita mempunyai category.js dengan data berupa berikut :

const categorySchema = new mongoose.Schema({

    category : {

        type : String,

        required : true,

    },

    images: [{type : mongoose.Schema.Types.ObjectId}],

    description: {

        type : String,

        required : true

    },

    image\_url : {

        type : String,

        required : true

    }

})

Image.js yang merupakan database dari foto yang telah diuplaod tersebut :

const imageSchema = new mongoose.Schema({

    title : {

        type : String,

        required : true,

    },

    source : {

        type : String,

        required : true,

    },

    detail : {

        description : String,

        raw : {

            megapixel : String,

            camera : String,

            iso : String,

            ss : String,

            aperture : String,

        }

    },

    author : {

        type : String,

        required : true,

        ref: "users"

    },

    liked\_by : [{

        type : mongoose.Schema.Types.ObjectId,

        ref : "users"

    }],

    collect\_by : [{

        type : mongoose.Schema.Types.ObjectId,

        ref : "users"

    }],

    views : 'Number',

    searchQuery : String,

    createdAt: {

        type: Date,

        default : Date.now,

    }

})

    required: true,

    unique: true,

  },

Terkahir adalah user.js untuk menampilkan database dari user baik dari password, username dan lain-lain

const userSchema = new mongoose.Schema({

    username : {

        type : String,

        required : true,

        unique: true

    },

    password : {

        type : String,

        required : true,

    },

    img\_profile : String,

    email : {

        type : String,

        required : true,

        unique : true

    },

    biography : String,

    location : String,

    website : String

})

Langkah kelima, kita akan membuat auth.js otetifikasi JWT untuk mengetahui setiap client sudah login atau belum dengan memeriksa cookie pada browser. Lalu, kita akan memasukan database yang dibutuhkan ke dalam route InsframeContent,js yang telah kita buat tersebut dengan database yang diperlukan seperti category membutuhkan data dari database category.js dan lain-lain. Lalu, kita akan membuat route untuk para user apabila user sudah login ke dalam website tersebut seperti edit profile dari masing-masing user tersebut , collection yang sudah ditambahkan kepada user, mengecek list foto yang telah diupload user, dan lain-lain. Intinya merupakan route yang berhubungan dengan user tersebut.

Langkah keenam, membuat route InsframeAuth.js untuk melakukan autentikasi apabila kita ingin meregister account atau log out dari user(menggunakan cookie) tersebut maka dibutuhkan autentikasi.

Langkah ketujuh, setelah website telah berjalan maka kita akan mendeploy website tersebut ke Heroku dengan membuat form application di Heroku yang telah kita buat dan hasil akhirnya bahwa website Insframe tersebut telah dideploy ke Heroku dan mendaptkan link sebagai berikut <https://insframephoto.herokuapp.com/>

|  |  |
| --- | --- |
| **Langkah** | **Aktivitas Anggota** |
| 1 | Melakukan instalisasi package yang dibutuhkan seperti mongoose, express dan lain-lain. Lalu, membuat server.js untuk server dari website tersebut. |
| 2 | Membuat route untuk mengambil data-data yang akan dilempar ke ejs. Route pertama yang dibuat adalah InsframeContent.js dan melakukan res.render ejs tersebut |
| 3 | Membuat ejs untuk tampilan front end website |
| 4 | Membuat database di MonggoDB untuk data-data user , image, dan lain-lain |
| 5 | Membuat autentikasi user apakah user sudah login di account tersebut yang berupa function yang dapat diambil dari route yang lain dan membuat route InsframeUser.js untuk melakukan route yang berhubungan dengan profile dari user tersebut. |
| 6 | Membuat autentikasi user apabila user ingin logut dari website atau ingin sign in untuk website tersebut. |
| 7 | Memasukan database yang diperlukan di semua route seperti category yang membutuhkan database category.js tersebut. |
| 8 | Melakukan deploy website ke heroku |