**Media Streaming with IBM Cloud Video Streaming**

**Phase – 3 Development Part 1**

**Problem Statement:**

Set up user registration and authentication mechanisms to ensure secure access to the platform.

**Platform Features:**

* + **Account Creation and Management**
  + **Chat Interface**
  + **User create Playlist**
  + **Recommendations**
  + **Fast and Responsive Layout**

**Goal: Design an intuitive user interface.**

Why to create an intuitive user interface is important. Some of the reasons are given below.

**User satisfaction:** Intuitive interfaces are easy to understand and navigate, leading to a positive user experience. Users can quickly accomplish tasks without feeling overwhelmed or confused, leading to higher satisfaction and retention rates.

**Reduced learning curve:** Intuitive interfaces simplify the learning process for new users, enabling them to grasp the functionality and features of a system more rapidly. This can lead to increased adoption rates and overall user engagement.

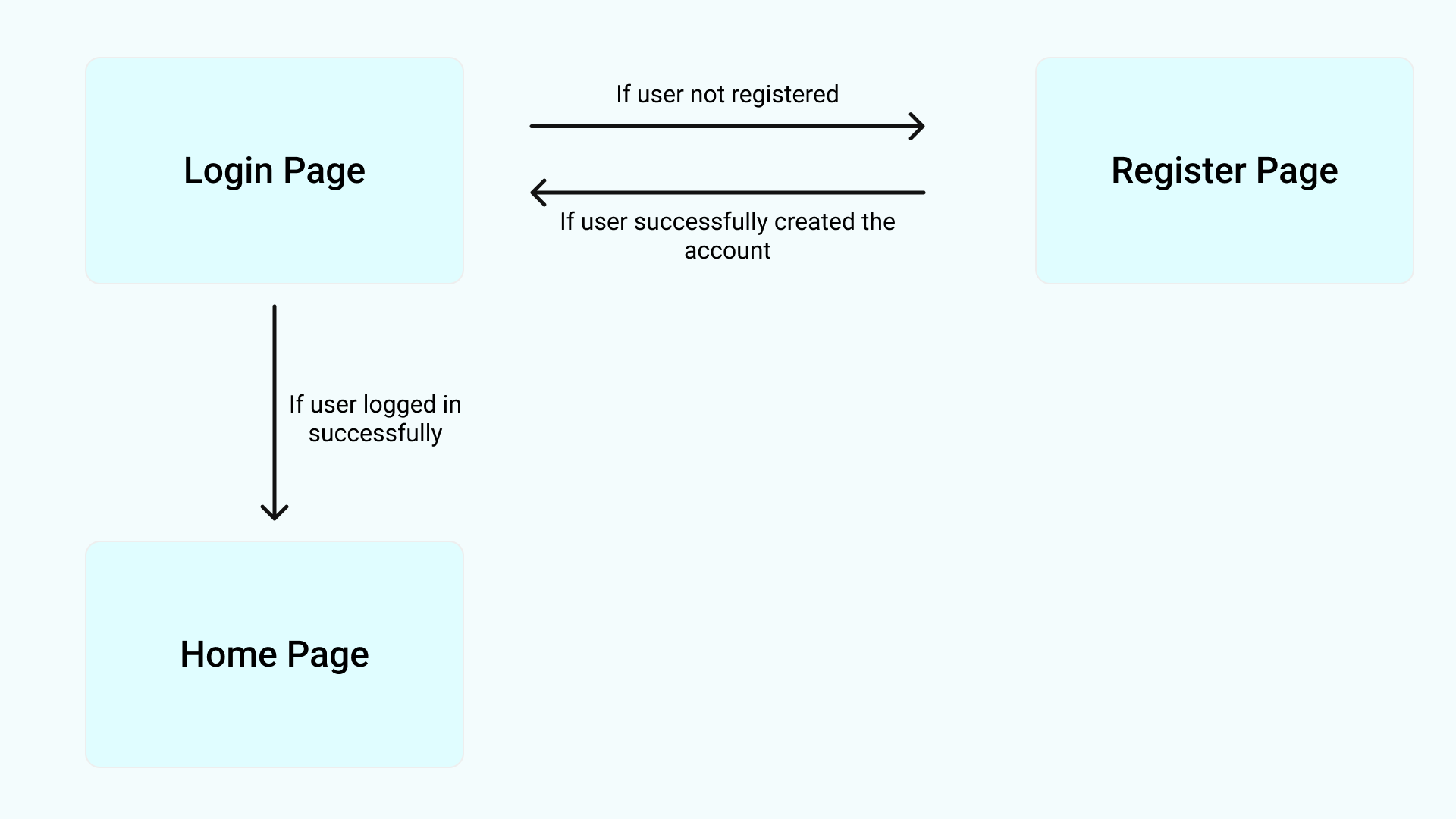
**Increased productivity:** Intuitive interfaces streamline workflows and minimize the time required to complete tasks. Users can focus on their objectives rather than spending unnecessary time trying to figure out how to use the interface, resulting in improved productivity.

**Decreased errors:** Intuitive interfaces reduce the likelihood of user errors and mistakes, as they guide users through processes and provide clear feedback. This can prevent frustration and the need for additional support or troubleshooting.

**Competitive advantage:** In today's market, user experience plays a significant role in the success of a product or service. An intuitive interface can differentiate a product from its competitors, attracting and retaining more users.

**Website Flow:**

Our website contains mainly three pages:

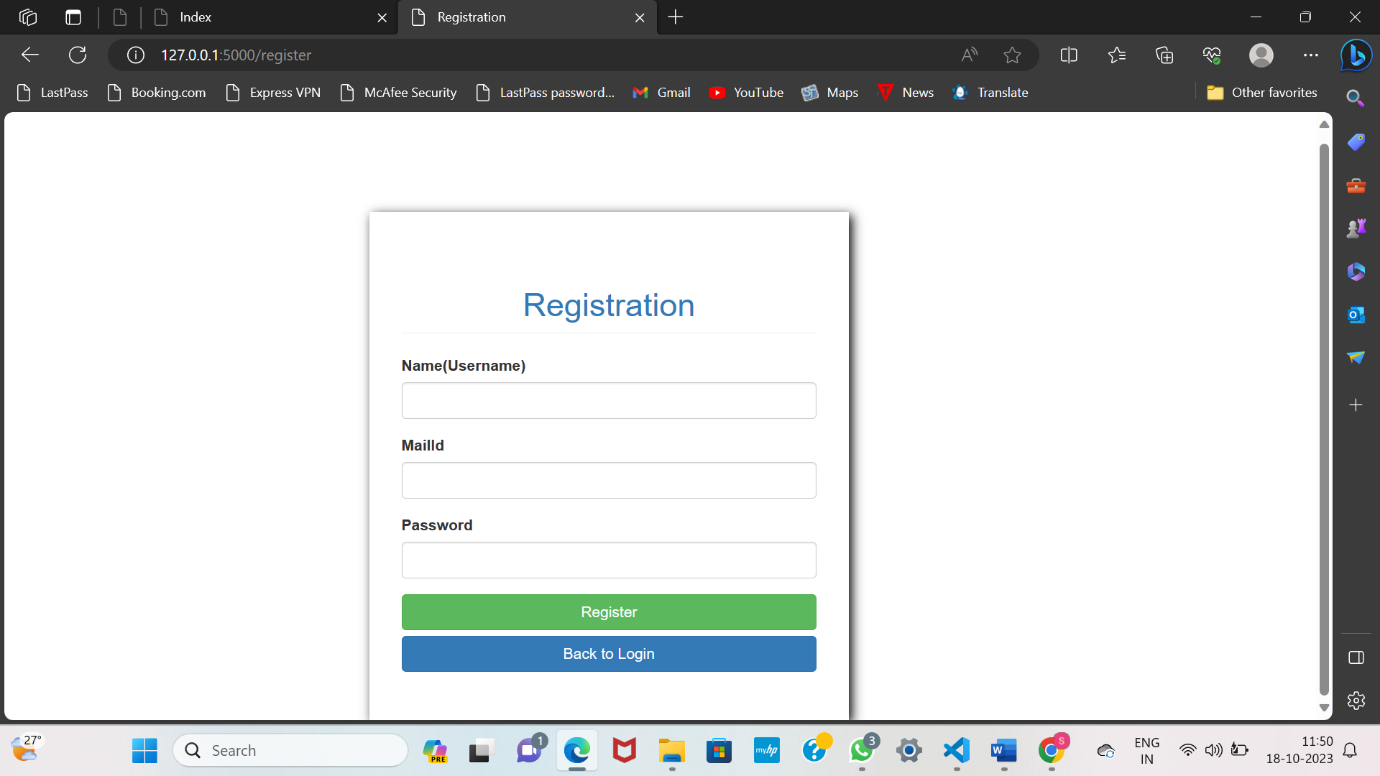


* + Login Page
  + Register Page
  + Home Page

**Account Creation and Management:**

User want to provide essential details like Username, Email Id,

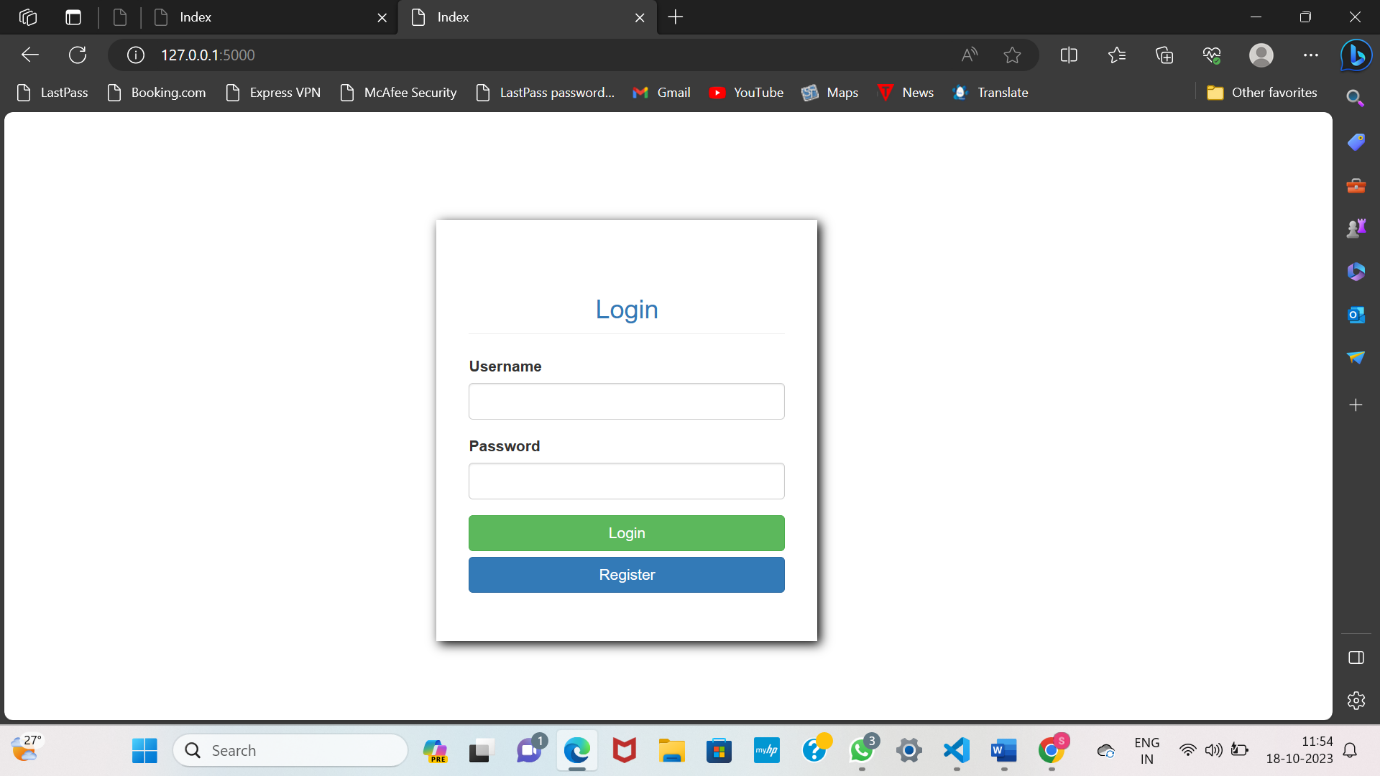
Password to register.



After successfully creating an account, user is automatically redirected to login page.

**Login Page:**

User wants to provide the essential account details like Username, Password to login successfully into his account.



After successfully logged in to his account, user home page will be displayed.

**Technology Stack Used:**

The technologies that we used to set up user registration and authentication mechanisms to ensure secure access to the platform are:

* + HTML
  + CSS
  + JavaScript
  + Flask
  + SQLite

**Flask and SQLite:**

Flask and SQLite are used in backend to enable users for creating account and managing it securely.

**HTML, CSS, JavaScript and other frameworks:**

Used to create an intuitive user interface with fast and responsive layout.

**Code which drives our backend:**

App.py

from flask import Flask,render\_template,request,flash,redirect,url\_for,session

import sqlite3

app = Flask(\_\_name\_\_)

app.secret\_key="123"

con=sqlite3.connect("database.db")

con.execute("create table if not exists customer(pid integer primary key,name text,address text,contact integer,mail text)")

con.close()

@app.route('/')

def index():

    return render\_template('index.html')

@app.route('/login',methods=["GET","POST"])

def login():

    if request.method=='POST':

        name=request.form['name']

        password=request.form['password']

        con=sqlite3.connect("database.db")

        con.row\_factory=sqlite3.Row

        cur=con.cursor()

        cur.execute("select \* from customer where name=? and mail=?",(name,password))

        data=cur.fetchone()

        if data:

            session["name"]=data["name"]

            session["mail"]=data["mail"]

            return redirect("customer")

        else:

            flash("Username and Password Mismatch","danger")

    return redirect(url\_for("index"))

@app.route('/customer',methods=["GET","POST"])

def customer():

    return render\_template("customer.html")

@app.route('/register',methods=['GET','POST'])

def register():

    if request.method=='POST':

        try:

            name=request.form['name']

            address= "null"

            mail= request.form['mail']

            password=request.form['password']

            con=sqlite3.connect("database.db")

            cur=con.cursor()

            cur.execute("insert into customer(name,address,contact,mail)values(?,?,?,?)",(name,address,mail,password))

            con.commit()

            flash("Record Added  Successfully","success")

        except:

            flash("Error in Insert Operation","danger")

        finally:

            return redirect(url\_for("index"))

            con.close()

    return render\_template('register.html')

@app.route('/logout')

def logout():

    session.clear()

    return redirect(url\_for("index"))

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

**Conclusion:**

In summary, the completion of phase 3 development part 1 has been instrumental in enhancing the user experience of our media streaming platform through the implementation of an intuitive and user-friendly interface. By prioritizing simplicity and ease of navigation, we aim to ensure that our users can seamlessly access and enjoy the content they desire. Additionally, the successful establishment of robust user registration and authentication mechanisms lays a solid foundation for ensuring the security and privacy of our users' information and content. As we move into the next phase, we are committed to further refining these features to provide a seamless, secure, and engaging media streaming experience for our valued users.