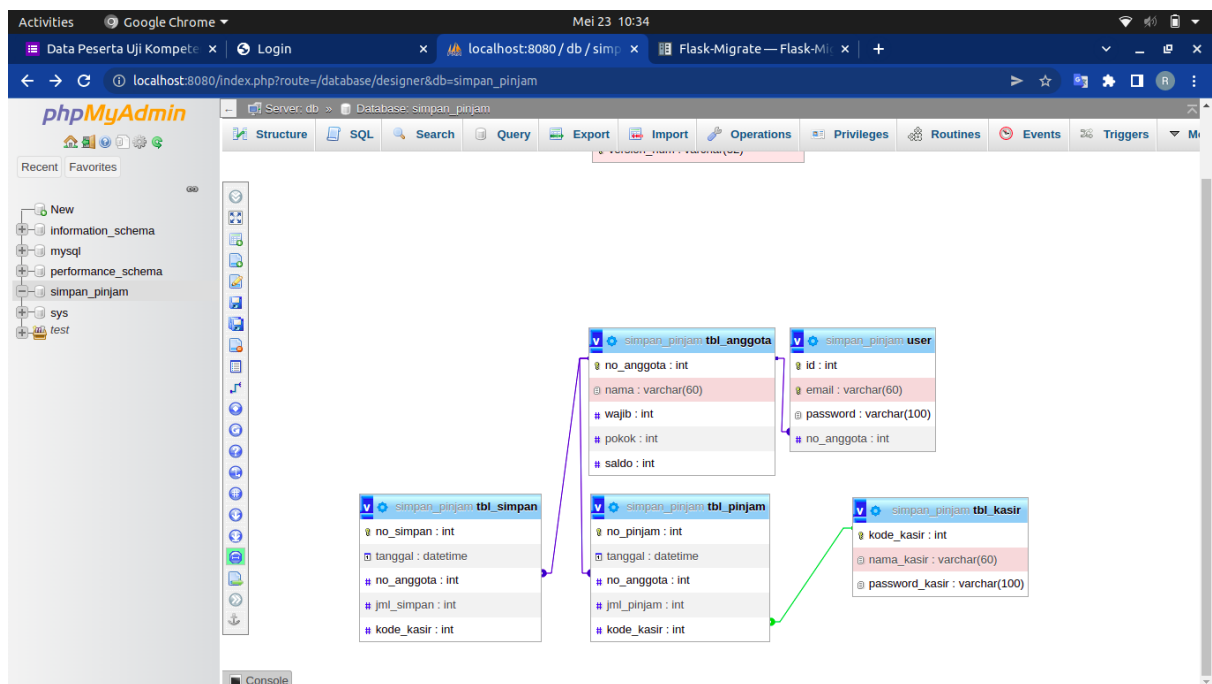


Rizky Darmadi -181011402252- Ujikom

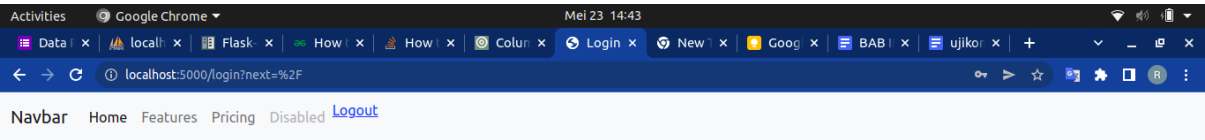
23 - Mei - 2022

Aplikasi Simpan Pinjam

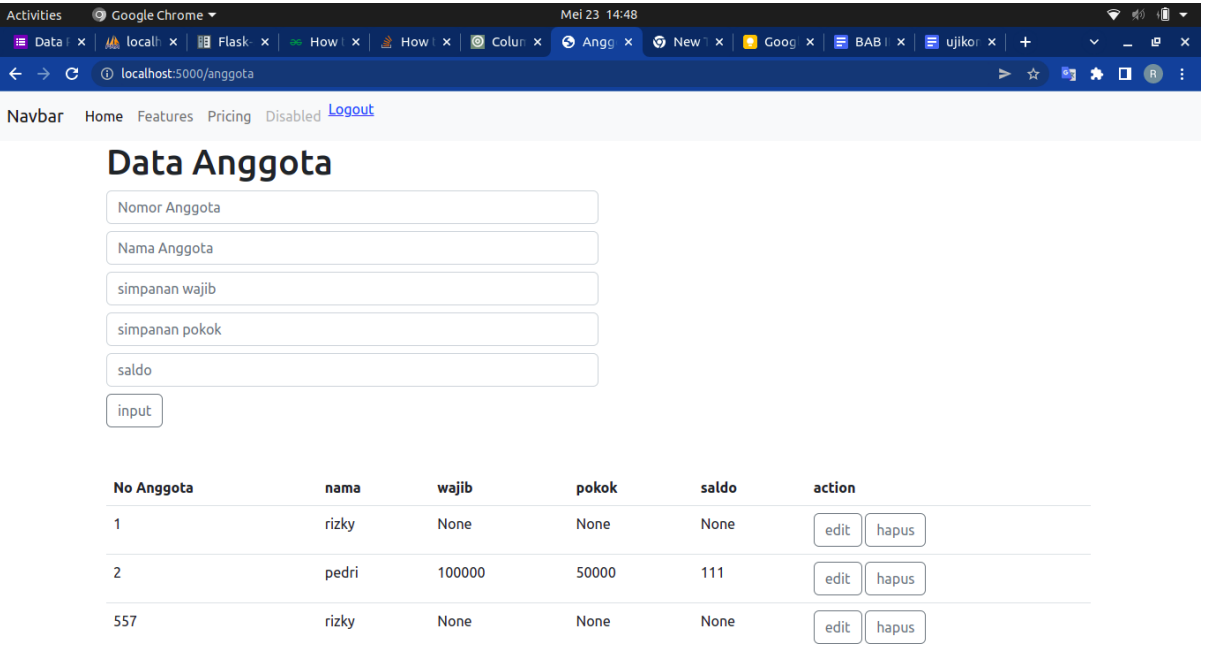
ERD



Halaman Awal



Halaman Setelah Login



Halaman Simpan

ActivitiesGoogle ChromeMei 23 14:51

Data xlocalh xFlask xHow xHow xColum xsimp: xNew xGoog xBAB l xujiko: x+ x

localhost:5000/simpan

NavbarHomeFeaturesPricingDisabledLogout

Data Simpan

No Simpan

No Anggota

Jumlah Simpan

input

No simpan	Tanggal	No Anggota	Jml Simpan	Kode Kasir	
2	2022-05-23 00:00:00	1	200	123	<div>edithapus</div>

Halaman pinjam

ActivitiesGoogle ChromeMei 23 14:54

Data xlocalh xFlask xHow xHow xColum xpinja: xNew xGoog xBAB l xujiko: x+ x

localhost:5000/pinjam

NavbarHomeFeaturesPricingDisabledLogout

Data Pinjam

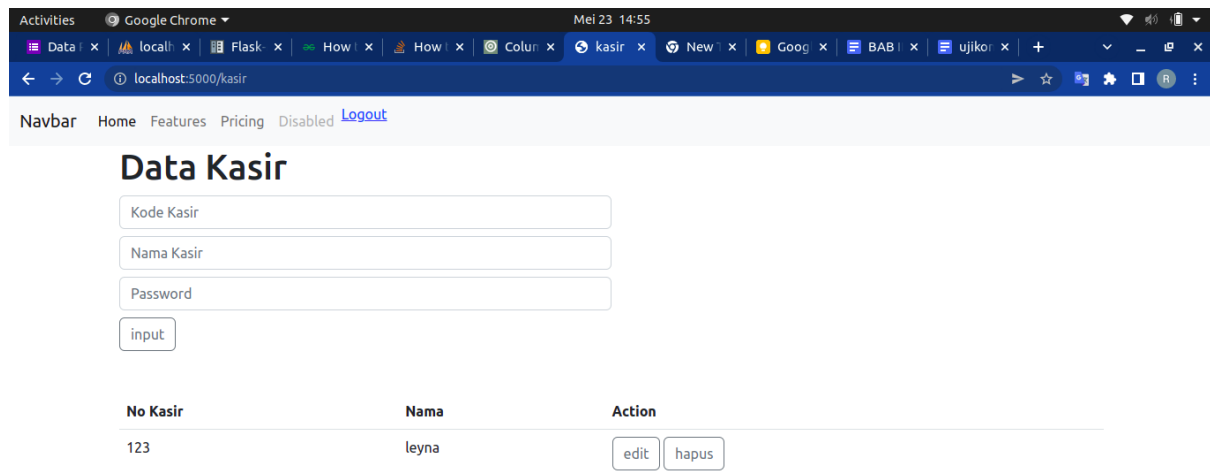
No pinjam

No Anggota

Jumlah pinjam

input

No pinjam	Tanggal	No Anggota	Jml pinjam	Kode Kasir	
11	2022-05-23 00:00:00	1	10	123	<div>edithapus</div>



Source Code

init.py

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from flask_login import LoginManager
from flask_migrate import Migrate

db = SQLAlchemy()
def create_app():
    app = Flask(__name__)
    app.config['SECRET_KEY'] = 'secret-key-goes-here'
    app.config['SQLALCHEMY_DATABASE_URI'] =
'mysql+pymysql://root:root@localhost:3308/simpan_pinjam'
    app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
    db.init_app(app)

    login_manager = LoginManager()
    login_manager.login_view = 'auth.login'
    login_manager.init_app(app)
    from .models import User
    @login_manager.user_loader
    def load_user(user_id):
        return User.query.get(int(user_id))
```

```

migrate = Migrate(app, db) # for migrate
https://flask-migrate.readthedocs.io/en/latest/

from .auth import auth as auth_blueprint
app.register_blueprint(auth_blueprint)
from .anggota import anggota as anggota_blueprint
app.register_blueprint(anggota_blueprint)
from .kasir import kasir as kasir_blueprint
app.register_blueprint(kasir_blueprint)
from .simpan import simpan as simpan_blueprint
app.register_blueprint(simpan_blueprint)
from .pinjam import pinjam as pinjam_blueprint
app.register_blueprint(pinjam_blueprint)
from .main import main as main_blueprint
app.register_blueprint(main_blueprint)
return app

```

auth.py

```

from flask import
Blueprint, request, render_template, redirect, url_for, flash
from .models import User, TBLAnggota
from werkzeug.security import generate_password_hash,
check_password_hash
from __init__ import db
from flask_login import login_user, logout_user, login_required

auth = Blueprint('auth', __name__)

i = 0
def mydefault():
    global i
    i += 1
    return i

@auth.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'GET':
        return render_template('login.html', title='Login')
    else:
        email = request.form.get('user')
        password = request.form.get('password')

        user_data = User.query.filter_by(email=email).first()

```

```

        if not user_data:
            flash('Please sign up before!')
            return redirect(url_for('auth.signup'))
        elif not check_password_hash(user_data.password, password):
            flash('Please check your login details and try again.')
        login_user(user_data)
        return redirect(url_for('main.index'))

@auth.route('/signup', methods=['GET', 'POST'])
def signup():
    if request.method=='GET':
        return render_template('signUp.html')
    else:
        email = request.form.get('email')
        name = request.form.get('name')
        password = request.form.get('password')
        no_anggota = request.form.get('no_anggota')
        confirm_password = request.form.get('confirm_password')

        if password != confirm_password:
            flash('Incorrect Password')
            return redirect(url_for('auth.signup'))

        user = User.query.filter_by(email=email).first()
        if user:
            flash('Email address already exists')
            return redirect(url_for('auth.signup'))

        new_anggota = TBLAnggota(nama=name)
        db.session.add(new_anggota)
        db.session.commit()

        new_user = User(email=email,
password=generate_password_hash(password,
method='sha256'),no_anggota=int(no_anggota))
        db.session.add(new_user)
        db.session.commit()

        return redirect(url_for('auth.login'))

@auth.route('/logout') # define logout path
@login_required

```

```
def logout(): #define the logout function
    logout_user()
    return redirect(url_for('main.index'))
```

kasir.py

```
from flask import
Blueprint, request, render_template, redirect, url_for, flash
from .models import TBLKasir
from .__init__ import db

kasir = Blueprint('kasir', __name__)

@kasir.route('/kasir', methods=['GET', 'POST'])
def add_kasir():
    if request.method=='GET':
        kasir = TBLKasir.query.all()
        return render_template('kasir.html', title='kasir', kasir=kasir)
    else:

        kode_kasir = request.form.get('kode_kasir')
        nama_kasir = request.form.get('nama_kasir')
        password_kasir = request.form.get('password_kasir')

        new_kasir =
TBLKasir(nama_kasir=nama_kasir, kode_kasir=kode_kasir, password_kasir=password_kasir)
        db.session.add(new_kasir)
        db.session.commit()

        return redirect(url_for('kasir.add_kasir'))

@kasir.route('/kasir/edit/<int:id>', methods=['GET', 'POST'])
def update_kasir(id):
    if request.method=='GET':
        return render_template('kasir_update.html', title='kasir')
    else:
        kasir = TBLKasir.query.filter_by(kode_kasir=id).first()

        kasir.nama_kasir = request.form.get('nama_kasir')
        kasir.password_kasir = int(request.form.get('password_kasir'))
        kasir.kode_kasir = int(id)

        db.session.commit()
```

```

        return redirect(url_for('kasir.add_kasir'))

@kasir.route('/kasir/hapus/<int:id>', methods=['GET', 'POST'])
def delete_kasir(id):

    kasir = TBLKasir.query.filter_by(kode_kasir=id).first()
    db.session.delete(kasir)
    db.session.commit()

    return redirect(url_for('kasir.add_kasir'))

```

pinjam.py

```

from flask import
Blueprint, request, render_template, redirect, url_for, flash
from .models import TBLPinjam, TBLPinjam
from .__init__ import db
from datetime import date

pinjam = Blueprint('pinjam', __name__)

@pinjam.route('/pinjam', methods=['GET', 'POST'])
def add_pinjam():
    if request.method == 'GET':
        pinjam = TBLPinjam.query.all()
        return
    render_template('pinjam.html', title='pinjam', pinjam=pinjam)
    else:

        no_pinjam = request.form.get('no_pinjam')
        no_anggota = request.form.get('no_anggota')
        jumlah_pinjam = request.form.get('jumlah_pinjam')

        new_pinjam =
TBLPinjam(no_pinjam=int(no_pinjam), no_anggota=int(no_anggota), jml_pinjam=int(jumlah_pinjam), tanggal=date.today(), kode_kasir=123)
        db.session.add(new_pinjam)
        db.session.commit()

```



```

        return redirect(url_for('pinjam.add_pinjam'))

@pinjam.route('/pinjam/edit/<int:id>', methods=['GET', 'POST'])
def update_pinjam(id):
    if request.method=='GET':
        return render_template('pinjam_update.html',title='pinjam')
    else:
        pinjam = TBLPinjam.query.filter_by(no_pinjam=id).first()

        pinjam.no_pinjam = int(request.form.get('no_pinjam'))
        pinjam.nama_pinjam = request.form.get('nama_pinjam')
        pinjam.pinjaman_wajib = int(request.form.get('pinjaman_wajib'))
        pinjam.pinjaman_pokok = int(request.form.get('pinjaman_pokok'))
        pinjam.saldo = int(request.form.get('saldo'))

        db.session.commit()

    return redirect(url_for('pinjam.add_pinjam'))

@pinjam.route('/pinjam/hapus/<int:id>', methods=['GET', 'POST'])
def delete_pinjam(id):

    pinjam = TBLPinjam.query.filter_by(no_pinjam=id).first()
    db.session.delete(pinjam)
    db.session.commit()

    return redirect(url_for('pinjam.add_pinjam'))

```

simpan.py

```

from flask import
Blueprint,request,render_template,redirect,url_for,flash
from .models import TBLSimpan
from .__init__ import db
from datetime import date

simpan = Blueprint('simpan', __name__)

@simpan.route('/simpan', methods=['GET', 'POST'])
def add_simpan():

```

```

    if request.method=='GET':
        simpan = TBLSimpan.query.all()
        return
render_template('simpan.html',title='simpan',simpan=simpan)
    else:

        no_simpan = request.form.get('no_simpan')
        no_anggota = request.form.get('no_anggota')
        jumlah_simpan = request.form.get('jumlah_simpan')

        new_simpan =
TBLSimpan(no_simpan=int(no_simpan),no_anggota=int(no_anggota),jml_simpan=int(jumlah_simpan),tanggal=date.today(),kode_kasir=123)
        db.session.add(new_simpan)
        db.session.commit()

        return redirect(url_for('simpan.add_simpan'))

@simpan.route('/simpan/edit/<int:id>', methods=['GET', 'POST'])
def update_simpan(id):
    if request.method=='GET':
        return render_template('simpan_update.html',title='simpan')
    else:
        simpan = TBLSimpan.query.filter_by(no_simpan=id).first()

        simpan.no_anggota = request.form.get('no_anggota')
        simpan.jml_simpan = int(request.form.get('jumlah_simpan'))

        db.session.commit()

        return redirect(url_for('simpan.add_simpan'))

@simpan.route('/simpan/hapus/<int:id>', methods=['GET', 'POST'])
def delete_simpan(id):

    simpan = TBLSimpan.query.filter_by(no_simpan=id).first()
    db.session.delete(simpan)
    db.session.commit()

    return redirect(url_for('simpan.add_simpan'))

```

