Project simplified database

Students
-id_student -name -email -phone -birthdate

Classes
id_class class_name id_user (teacher) date time

Attendence
Id Id_class Id_students Attend(bool)

Users (more that 1 is teacher)
id_user name login password .....

You can add more fields(columns) to the tables.

To create the database, either use sql to create, or use the models with create in the end.

Model - Simplified SQLite Schema

Create a empty database, and after run this SQL to create tables

```
-- USERS (admins and teachers)
CREATE TABLE users (
    id_user INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL,
    login TEXT UNIQUE NOT NULL,
    password TEXT NOT NULL
);

-- STUDENTS
CREATE TABLE students (
    id_student INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL,
    email TEXT UNIQUE NOT NULL,
    phone TEXT,
    birthdate TEXT  -- ISO format: YYYY-MM-DD
);

-- CLASSES
CREATE TABLE classes (
    id_class INTEGER PRIMARY KEY AUTOINCREMENT,
    class_name TEXT NOT NULL,
    id_user INTEGER NOT NULL,  -- Teacher
    date TEXT NOT NULL,        -- ISO format
    time TEXT NOT NULL,        -- HH:MM format
    FOREIGN KEY (id_user) REFERENCES users(id_user)
);

-- ATTENDANCE
CREATE TABLE attendance (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    id_class INTEGER NOT NULL,
    id_student INTEGER NOT NULL,
    attend INTEGER NOT NULL CHECK(attend IN (0,1)),
    FOREIGN KEY (id_class) REFERENCES classes(id_class),
    FOREIGN KEY (id_student) REFERENCES students(id_student),
    UNIQUE (id_class, id_student)  -- One record per student per class
);
```

Run this for inserting data

```sql
-- USERS: 1 Admin, 2 Teachers
INSERT INTO users (name, login, password) VALUES
  ('Alice Admin', 'admin', 'admin123'),
  ('Tom Teacher', 'tom', 'teach123'),
  ('Jane Teacher', 'jane', 'teach456');

-- STUDENTS
INSERT INTO students (name, email, phone, birthdate) VALUES
  ('John Doe', 'john@example.com', '123456789', '2005-04-15'),
  ('Jane Roe', 'jane@example.com', '987654321', '2006-08-22');

-- CLASSES
INSERT INTO classes (class_name, id_user, date, time) VALUES
  ('Math 101', 2, '2025-06-01', '09:00'),
  ('Science Basics', 3, '2025-06-02', '10:30');

-- ATTENDANCE
INSERT INTO attendance (id_class, id_student, attend) VALUES
  (1, 1, 1),  -- John present in Math 101
  (1, 2, 0),  -- Jane absent in Math 101
  (2, 2, 1);  -- Jane present in Science Basics
```

models.py (Flask + Peewee)

This is models.py to access the database (required). if you use the create database (in the end) it will create the database also. Is not, it just accesses the database tables.

```python
from peewee import *
from datetime import date

# SQLite database
db = SqliteDatabase('classes.db')

# Base model
class BaseModel(Model):
    class Meta:
        database = db

# Users: Admins and Teachers
class User(BaseModel):
    id_user = AutoField()
    name = CharField()
    login = CharField(unique=True)
    password = CharField()  # Store hashed passwords in production

# Students
class Student(BaseModel):
    id_student = AutoField()
    name = CharField()
    email = CharField(unique=True)
    phone = CharField(null=True)
    birthdate = DateField()

# Classes (one teacher per class)
class Class(BaseModel):
    id_class = AutoField()
    class_name = CharField()
    id_user = ForeignKeyField(User, backref='classes')
    date = DateField()
    time = TimeField()

# Attendance
class Attendance(BaseModel):
    id = AutoField()
    id_class = ForeignKeyField(Class, backref='attendances')
    id_student = ForeignKeyField(Student, backref='attendances')
    attend = BooleanField()  # True if present, False if absent

    class Meta:
        indexes = (
            (('id_class', 'id_student'), True),  # Unique constraint
        )

# Create tables
def initialize_db():
    with db:
        db.create_tables([User, Student, Class, Attendance])
```

Usage Example

Add this to your main Flask app or an initialization script:

```
from models import initialize_db

if __name__ == '__main__':
    initialize_db()
    print("Database tables created.")
```