**Database Details**

**Database used: sqlite**

We have used sqlite database indirectly. Indiretly because in Flask there is one library

called 'Flask-SQLAlchemy' which connect python classes to the sqlite database.

Flask-SQLAlchemy uses Object Relational Mapper (ORM).

ORM basically is a Object oriented way of dealing with databases.

i. Tables as a Classes

ii. Fields/Columns as a Arguments

In models.py we have used the two main model for this project and they are User and Post.

You can see they are written as a python class. It includes column and fields. But sqlite

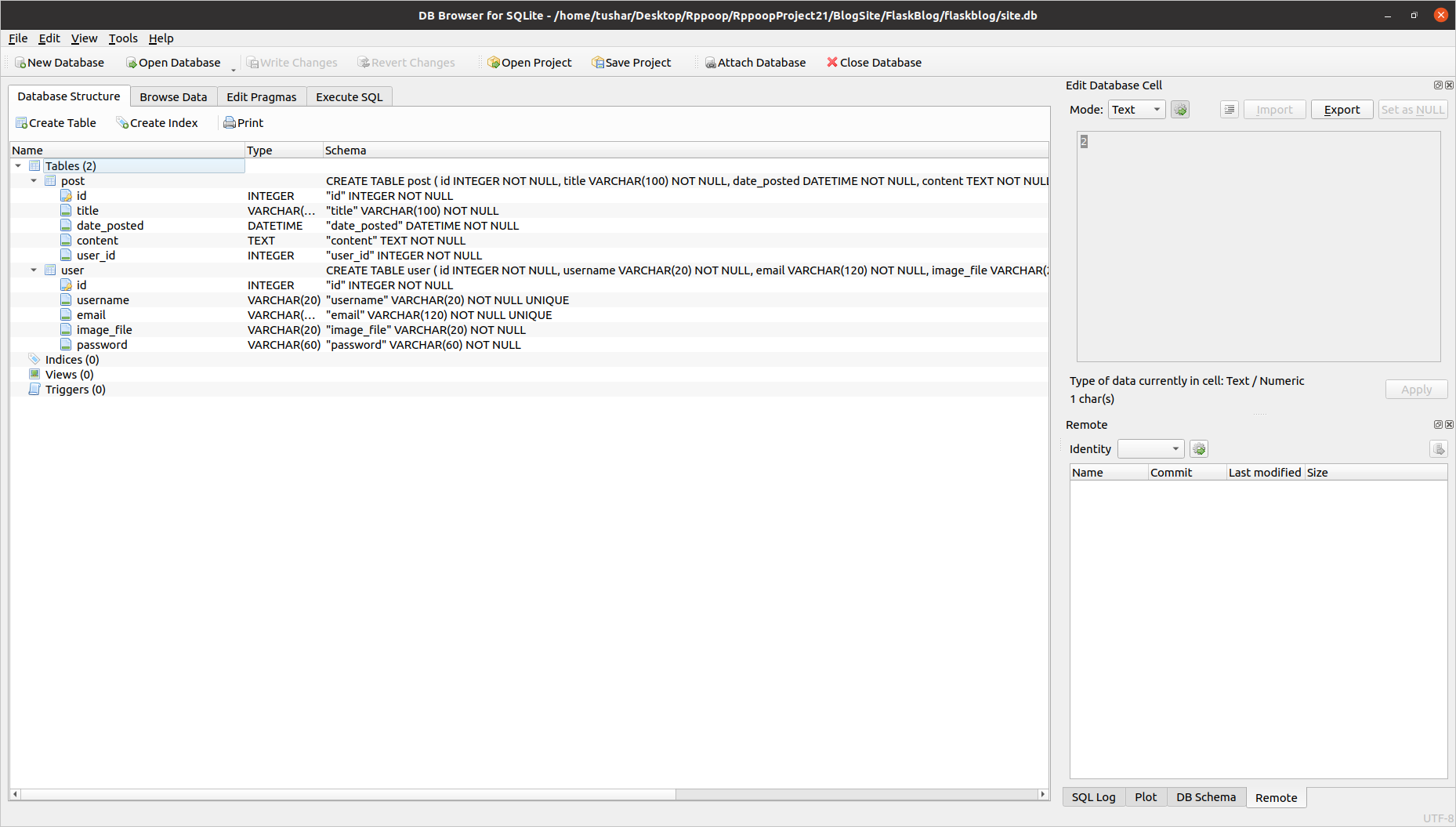
database includes Table and Column like structure. So SQLAlchemy take care of conversion

of classes and fields into Table and Columns.

In addition to this, we have used SQL which is querying language to access database data

in project wherever required.

We have created two models in models.py for database and they are user and post.



**1. User Model/class:**

**class User(db.Model, UserMixin):**

**id = db.Column(db.Integer, primary\_key=True)**

**username = db.Column(db.String(20), unique=True, nullable=False)**

**email = db.Column(db.String(120), unique=True, nullable=False)**

**image\_file = db.Column(db.String(20), nullable=False, default='default.png')**

**password = db.Column(db.String(60), nullable=False)**

**posts = db.relationship('Post', backref='author', lazy=True)**

For above User class, one table named 'user' is created in the database.

User class has 5 attributes (id, username, email, image\_file and password), so in

the table created, total 5 Columns are created with name of column same as that

of class attributes.

* **id**: It stores the id for each user created. It is a primary key means we

can access every details of a user using it's id.

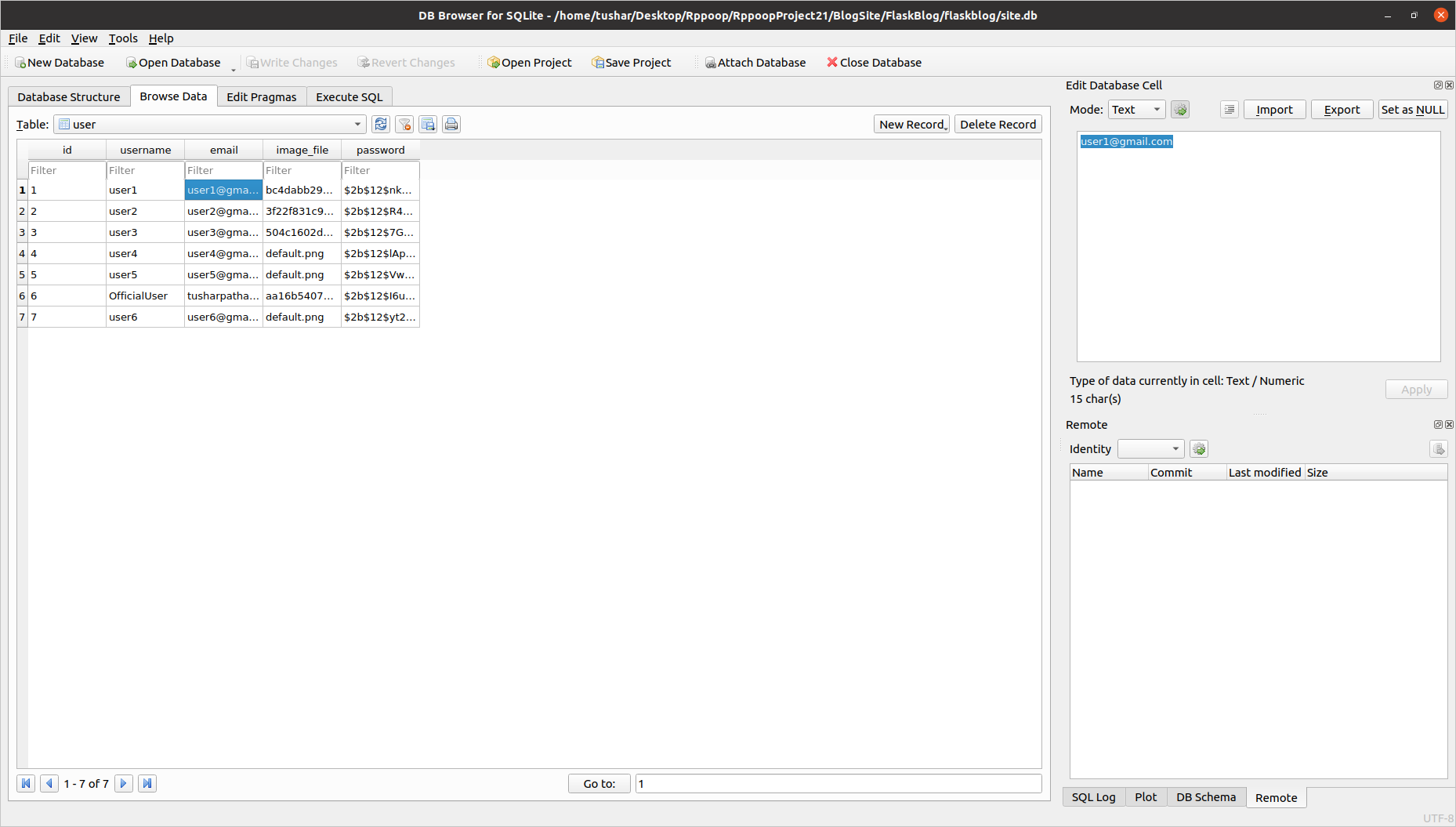
* **username**: It stores the username of a user.
* **email**: It stores the email addess of a user.
* **image\_file**: It stores tha name of profile image file. Instead of storing the image

with actual filename we have given random token as name generated

using secrets module to avoid common name.

* **password**: It stores tha password of a user. First password is encrypted using

Bcrypt python library and then they are being stored into the database.



Note: No column will be created for last attribute 'posts' as it is defining relationship

with other class/table here it is 'post'. We can access post table data via posts attribute

of user class.

**2. Post Model/class:**

**class Post(db.Model):**

**id = db.Column(db.Integer, primary\_key=True)**

**title = db.Column(db.String(100), nullable=False)**

**date\_posted = db.Column(db.DateTime, nullable=False, default=datetime.utcnow)**

**content = db.Column(db.Text, nullable=False)**

**user\_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)**

For Post class also, one table named 'post' is created in the database.

As it has 5 attributes (id, title, date\_posted, content, user\_id), so 5 column with same

name as attributes are created in 'post' table.

* **id**: It stores the id for each post. It is primary key for table Post.
* **title**: It stores the title of a post.
* **date\_posted**: It stores the date and time of post.
* **content**: It stores the all content of a post.
* **user\_id**: It stores the id of an user to which this post belongs to.

