NAME: VISHWAKARMA MANISHKUMAR DURBHASH

REGNO.19BCE10303 CAMPUS: VIT BHOPAL

EMAIL: mdv932453@gmail.com

Phone: 8828071175

Task details: Instagram Backend API.

- Whole project is done with the help of standard libraries only no external libraries are used.
- All the APIs are successfully implemented.
- All the APIs are thread safe
- Passwords are encrypted.
- MongoDB is used for storage.

How to run this project?

- Download Golang , MongoDB, MongoDB drivers
- Create Database with name Instagram
- Create collections Users and Posts
- Start the server by running main.go file using command : go run main.go

Note:

The server will listen to localhost:9094 (as wriiten in program)

The Mongodb database will run on localhost:27017

Visit the following url to see the endpoints:

1. Create an User

URL: http://localhost:9094/users

Request Parameter	Data Type	Status	Remark
id	String	Mandatory	Method= **POST**
name	String	Mandatory	
email	String	Mandatory	
password	String	Mandatory	

Request sample:

{

"id": "125",

"name": "manish vishwakarma",

"email": "mdv932453@gmail.com",

"password": "Manish@2000"

}

Response sample: status 200 ok

In database:

{

{"_id": "$oid":"61615525b9cb7a9bcdb0100d"

"id": "125",

"name": "manish vishwakarma",

"email": "mdv932453@gmail.com",

"password":"hijọ�G▯ ^L�t��w��R▯ ��CT������\n�R�;�◌̇b�2"

}

Note: (Passwords are stored Encrypted)

2. Get a user using id

Method = **GET**
URL: http://localhost:9094/users/ <id here>
Sample URL: http://localhost:9094/users/125
Response Sample:

```
{

   "id": "125",

   "name": "manish vishwakarma",

   "email": "mdv932453@gmail.com",

   "password": "Manish@2000"

}
```

Note: (Password is decrypted and then returned)

3. Create a Post

URL: http://localhost:9094/posts

Request Parameter	Data Type	Status	Remark
id	String	Mandatory	Method= **POST**
caption	String	Mandatory	
image_url	String	Mandatory	
timestamp			

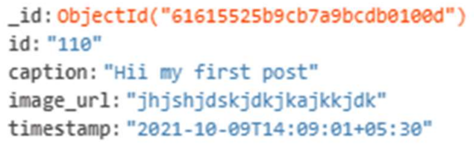
Note: Timestamp is automatically stored

Request sample:

```
{
    "id": "110",
    "caption": "Hii my first post",
    "image_url": "jhjshjdskjdkjkajkkjdk"
}
```

Response sample: status 200 ok

In database:



```
_id: ObjectId("61615525b9cb7a9bcdb0100d")
id: "110"
caption: "Hii my first post"
image_url: "jhjshjdskjdkjkajkkjdk"
timestamp: "2021-10-09T14:09:01+05:30"
```

4. Get a post using id

Method = **GET**
URL: http://localhost:9094/posts/ <id here>
Sample URL: http://localhost:9094/posts/110

Response sample:

```
{
    "id": "115",
    "caption": "Hii my first post",
    "image_url": "jhjshjdskjdkjkajkkjdk",
    "timestamp": "2021-10-09T14:09:01+05:30"
}
```

5. List all posts of a user

Method = **GET**
URL: http://localhost:9094/posts/users/ <id here>
Sample URL: http://localhost:9094/posts/users/110

Response sample:

```
[

  {

    "id": "110",

    "caption": "Hii my first post",

    "image_url": "jhjshjdskjdkjkajkkjdk",

    "timestamp": "2021-10-09T14:09:01+05:30"

  },
{
   "id": "110",
   "caption": "Hii my second post",
   "image_url": "jhjshjdskjdkjkajkkjdk",
   "timestamp": "2021-10-09T18:57:17+05:30"
},
{
   "id": "110",
   "caption": "Hii my third post",
   "image_url": "jhjshjdskjdkjkajkkjdk",
   "timestamp": "2021-10-09T18:58:17+05:30"
},
{
   "id": "110",
   "caption": "Hii my fourth post",
   "image_url": "jhjshjdskjdkjkajkkjdk",
   "timestamp": "2021-10-09T18:58:26+05:30"
},
{
   "id": "110",
   "caption": "Hii my fifth post",
   "image_url": "jhjshjdskjdkjkajkkjdk",
   "timestamp": "2021-10-09T18:58:34+05:30"
}
```

```
]
```

(Assumed that post id and user id are same since post can be uniquely identified by using "_id" parameter in mongoDB)