

Milestone 5

Joshua Nguyen
Lauren McIntire
Anthony Keydel
Kevin Brown-Hartland
Jacob Hans
Sasha Farhat

Team One (The Best)

User Acceptance Tests				
Step	Test step	Expected Result	Actual Result	Pass/fail
1	Test setting up the application and getting HTTP request	Status code of 200	Status code of 200	Pass
2	Wrong login. Enter user not in database	Flash 'wrong password'	Entered this part of code from coverage	Pass
3	Test User class to enter an new user	User.username = 'admin' User.password = 'default'	User.username = 'admin' User.password = 'default'	Pass
4	Test creation of server	Returns '<h1>Sent!</h1>'	Returns '<h1>Sent!</h1>'	Pass

Run 'make init'
Run 'make test'
Run 'make coverage'

```

py.test --cov-report term-missing --cov=tests
===== test session starts =====
platform linux2 -- Python 2.7.12, pytest-3.5.0, py-1.5.3, pluggy-0.6.0
rootdir: /home/user/Desktop/TheBest/Hartlands_Chat, inifile:
plugins: cov-2.5.1
collected 6 items

tests/test_app.py ..... [100%]

----- coverage: platform linux2, python 2.7.12-final-0 -----
Name                               Stmts   Miss  Cover   Missing
-----
tests/app.py                        45      10    78%    20, 34, 46-47, 51-54, 58-61, 77
tests/dummy.py                     15      15     0%    1-24
tests/tabledef.py                   16       0   100%
tests/test_app.py                   29       0   100%
-----
TOTAL                               105      25    76%

===== 6 passed in 0.30 seconds =====
user@cu-cs-vm:~/Desktop/TheBest/Hartlands_Chat$

```

Coverage for **tests/tabledef.py** : 100%



16 statements 16 run 0 missing 0 excluded

```

1 from sqlalchemy import *
2 from sqlalchemy import create_engine, ForeignKey
3 from sqlalchemy import Column, Date, Integer, String
4 from sqlalchemy.ext.declarative import declarative_base
5 from sqlalchemy.orm import relationship, backref
6
7 engine = create_engine('sqlite:///tutorial.db', echo=True)
8 Base = declarative_base()
9
10 #####
11 class User(Base):
12     """
13     __tablename__ = "users"
14
15     id = Column(Integer, primary_key=True)
16     username = Column(String)
17     password = Column(String)
18
19     #-----
20     def __init__(self, username, password):
21         """
22         self.username = username
23         self.password = password
24
25

```

```

19     #-----
20     def __init__(self, username, password):
21         """
22         self.username = username
23         self.password = password
24
25     # create tables
26     Base.metadata.create_all(engine)

```

Coverage for tests/app.py : 78%



45 statements 35 run 10 missing 0 excluded

```

1  from flask import Flask
2  from flask import Flask, flash, redirect, render_template, request, session, abort
3  import os
4  from sqlalchemy.orm import sessionmaker
5  from tabledef import *
6  from flask_socketio import SocketIO, send, emit
7
8  app = Flask(__name__)
9  app.config['SECRET_KEY'] = 'secretkey'
10 app.config['DEBUG'] = True
11 socketio = SocketIO(app)
12 engine = create_engine('sqlite:///tutorial.db', echo=True)
13 users = {}
14 placeholder = 'no-user'
15 @app.route('/')
16 def home():
17     if not session.get('logged_in'):
18         return render_template('login.html')
19     else:
20         return render_template('index.html')
21 @app.route('/login', methods=['POST'])
22 def do_admin_login():
23
24     POST_USERNAME = str(request.form['username'])
25     POST_PASSWORD = str(request.form['password'])
26
27     Session = sessionmaker(bind=engine)
28     s = Session()
29     query = s.query(User).filter(User.username.in_(POST_USERNAME), User.password.in_(POST_PASSWORD))
30     print("user is", query)
31     result = query.first()
32     print(result)
33     if result:
34         session['logged_in'] = True
35     else:
36         flash('wrong password!')
37     return home()
38

```

```

22 def do_admin_login():
23
24     POST_USERNAME = str(request.form['username'])
25     POST_PASSWORD = str(request.form['password'])
26
27     Session = sessionmaker(bind=engine)
28     s = Session()
29     query = s.query(User).filter(User.username.in_([POST_USERNAME]), User.password.in_([POST_PASSWORD]))
30     print("user is", query)
31     result = query.first()
32     print(result)
33     if result:
34         session['logged_in'] = True
35     else:
36         flash('wrong password!')
37     return home()
38
39 @app.route('/orginate')
40 def orginate():
41     socketio.emit('server orginated', 'Something happened on the server!')
42     return '<h1>Sent!</h1>'

```

Created with help from:

<https://damyanon.net/post/flask-series-testing/>
<http://flask.pocoo.org/docs/0.12/testing/>