Milestone 5

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

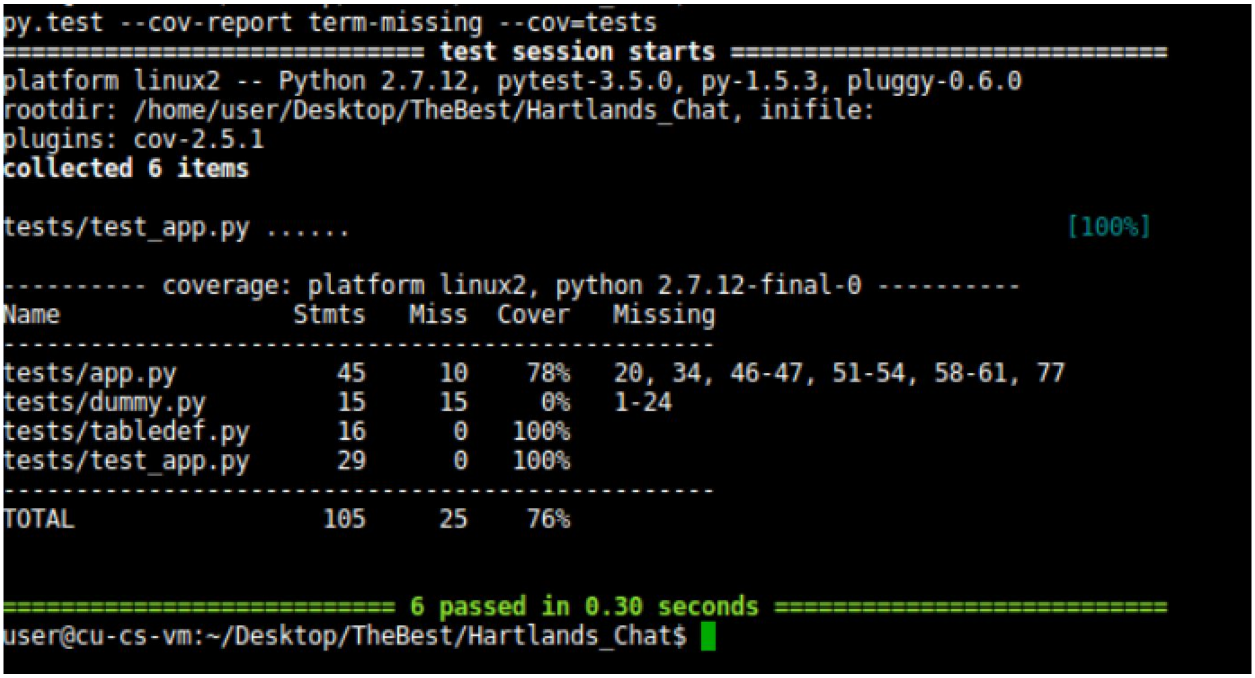
Team One (The Best)

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

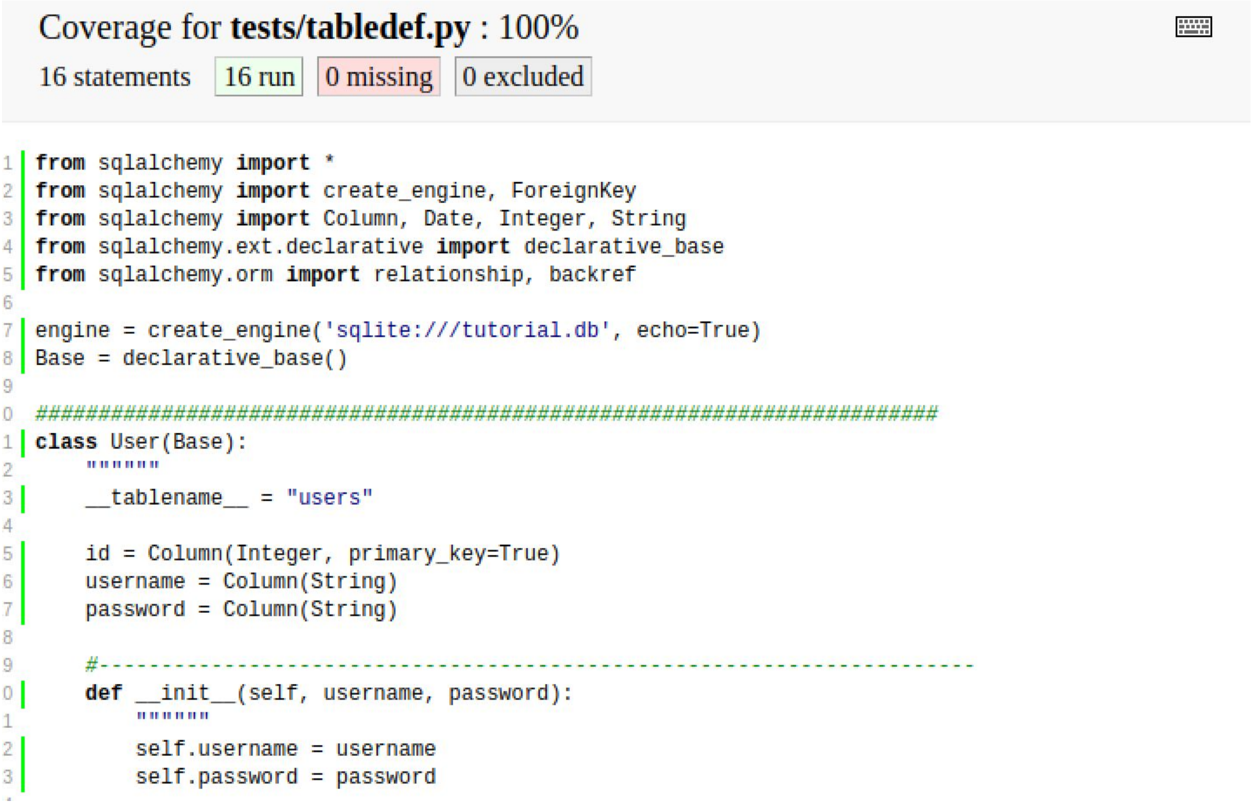
3	Test User class to enter an new user	User.username = ‘admin’
		User.password = ‘default’
			User.username = ‘admin’
			User.password = ‘default’
				Pass

4	Test creation of server	Returns ‘<h1>Sent!</h1>’	Returns ‘<h1>Sent!</h1>’	Pass

Run ‘make init’
Run ‘make test’
Run ‘make coverage’



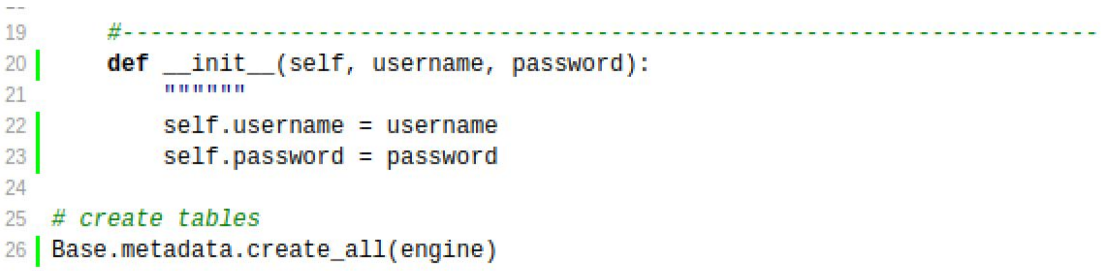
```
py.test --cov-report term-missing --cov=tests
============================= test session starts ==============================
platform linux2 -- Python 2.7.12, pytest-3.5.0, py-1.5.3, pluggy-0.6.0
rootdir: /home/user/Desktop/TheBest/Hartlands_Chat, inifile:
plugins: cov-2.5.1
collected 6 items

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$
```



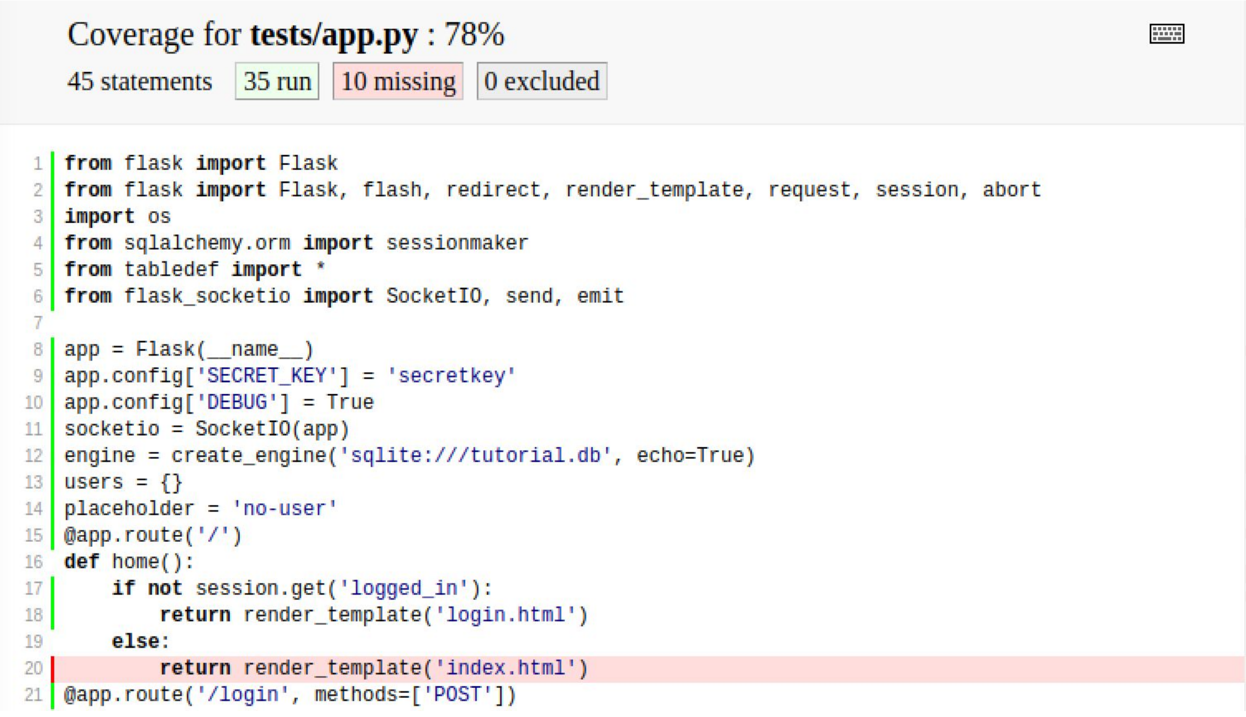
Coverage for **tests/tabledef.py** : 100%

16 statements 16 run 0 missing 0 excluded

```
from sqlalchemy import *
from sqlalchemy import create_engine, ForeignKey
from sqlalchemy import Column, Date, Integer, String
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import relationship, backref

engine = create_engine('sqlite:///tutorial.db', echo=True)
Base = declarative_base()

########################################################################
class User(Base):
    """"""
    __tablename__ = "users"

    id = Column(Integer, primary_key=True)
    username = Column(String)
    password = Column(String)

    #----------------------------------------------------------------------
    def __init__(self, username, password):
        """"""
        self.username = username
        self.password = password
```



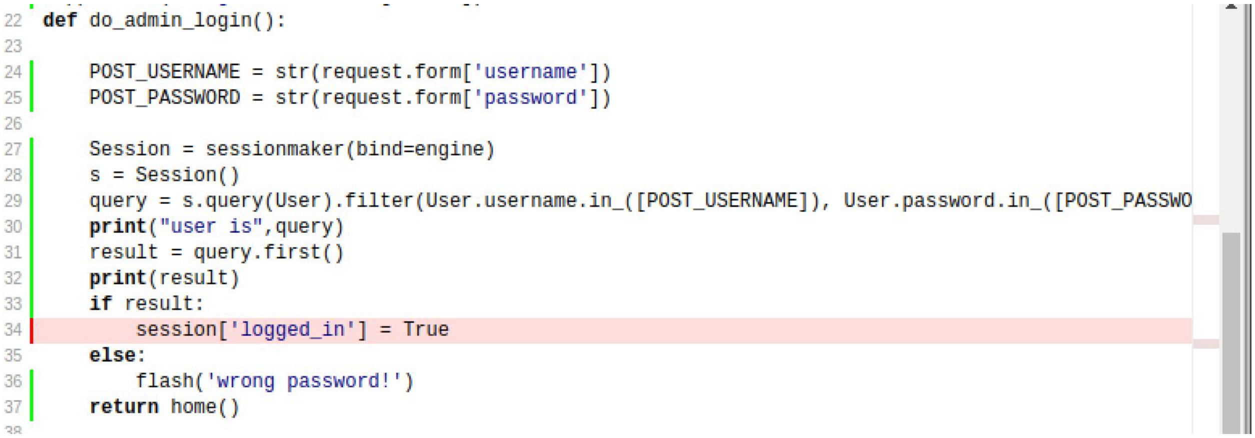
```
    #-------------------------------------------------------------------
    def __init__(self, username, password):
        """"""
        self.username = username
        self.password = password

# create tables
Base.metadata.create_all(engine)
```



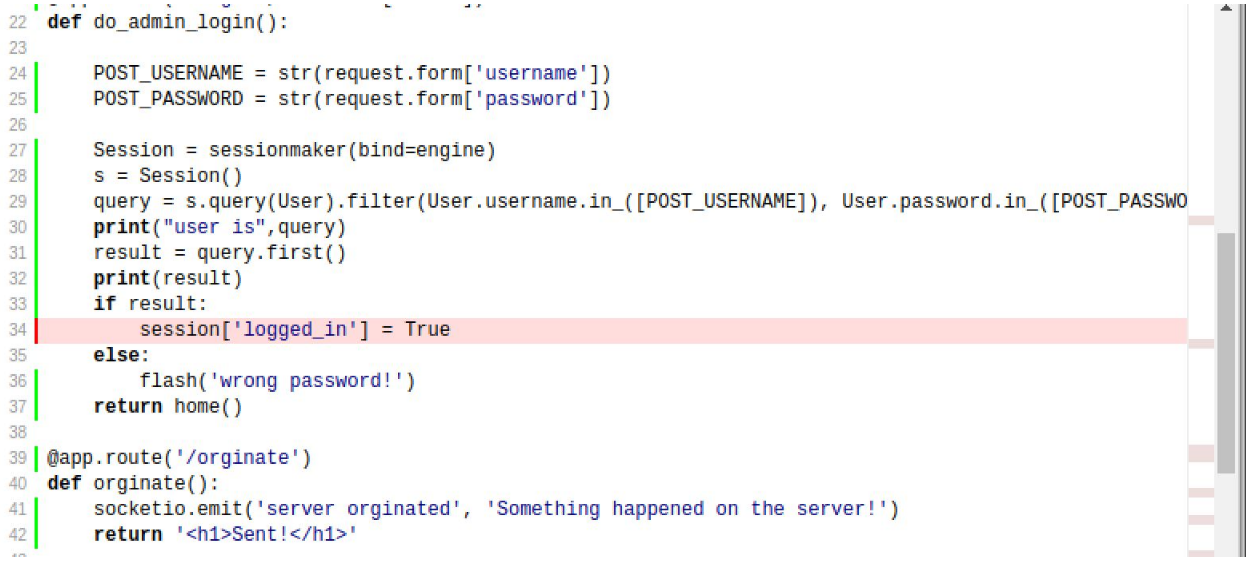
Coverage for **tests/app.py** : 78%

45 statements 35 run 10 missing 0 excluded

```
from flask import Flask
from flask import Flask, flash, redirect, render_template, request, session, abort
import os
from sqlalchemy.orm import sessionmaker
from tabledef import *
from flask_socketio import SocketIO, send, emit

app = Flask(__name__)
app.config['SECRET_KEY'] = 'secretkey'
app.config['DEBUG'] = True
socketio = SocketIO(app)
engine = create_engine('sqlite:///tutorial.db', echo=True)
users = {}
placeholder = 'no-user'
@app.route('/')
def home():
    if not session.get('logged_in'):
        return render_template('login.html')
    else:
        return render_template('index.html')
@app.route('/login', methods=['POST'])
```



```
def do_admin_login():

    POST_USERNAME = str(request.form['username'])
    POST_PASSWORD = str(request.form['password'])

    Session = sessionmaker(bind=engine)
    s = Session()
    query = s.query(User).filter(User.username.in_([POST_USERNAME]), User.password.in_([POST_PASSWO
    print("user is",query)
    result = query.first()
    print(result)
    if result:
        session['logged_in'] = True
    else:
        flash('wrong password!')
    return home()
```



```
def do_admin_login():

    POST_USERNAME = str(request.form['username'])
    POST_PASSWORD = str(request.form['password'])

    Session = sessionmaker(bind=engine)
    s = Session()
    query = s.query(User).filter(User.username.in_([POST_USERNAME]), User.password.in_([POST_PASSWO
    print("user is",query)
    result = query.first()
    print(result)
    if result:
        session['logged_in'] = True
    else:
        flash('wrong password!')
    return home()

@app.route('/orginate')
def orginate():
    socketio.emit('server orginated', 'Something happened on the server!')
    return '<h1>Sent!</h1>'
```