3 CITS3403/5505

1. (5 marks)

(3 Marks) Explain the key differences between agile software processes and any traditional software process.

A - Short iterations B - Design - Test- Code - Refactor C - People to People Client in the loop. D - Prototypos over docomentation F - others ...

• (2 Marks) Explain the difference between HTTP and HTML.

A - Hypertext Transfer Probocol 12 is a postocol for request and reparse packages to be transfer according to the principles of REST. 2 R C-Hyper Text Markup Longage In 13 langage for encoding veb page that is interpetable via Bransers In

4 CITS3403/5505

2. (10 marks) Describe, as clearly as you can, the appearence and style of the page generated by the attached code, when it is rendered by a modern web browser.

⊲html⊳ <meta charset='utf-8'/> <title>10 marks!</title> body {color: black; background-color: yellow;} .a {margin:1em; border:solid; padding:2px; background-color: white;} ol.b {list-style-type: lower-roman;} ul ul li + li{color: □blue;} ul > li.h:hover {color: _green} <h1>Agile Wed Dev Exam, Question 2</ $k_{
m V}$ <div class='a'> > An ordered list: ZoomTeamsSlack An unordered list: Examsoft/ExamplifyLMScssubmit </body> </div> ,11 Feet _ co l 🌽 Agile Wed Dev Exam, Question 2 C bander (l) • An ordered list: lem カi. Zoom ii. Teams rhite (P) iii. Slack Zpi iv. Github An unordered list: Examsoft/Examplify • LMS cssubmit blue Cost frot. (h) VI content

c1

a (

 $\begin{array}{c} 6 \\ \mathrm{CITS3403}/5505 \end{array}$

3. (10 marks) Write an HTML page, including javascript, to have the following functiontionality:

- (a) The page should have a heading, "My To Do List".
- (b) The page has an unordered list of items, the to-do list. 1
- (c) There is a text box, where new items can be written, with a button labelled "Add" beside it. 1/2
- (d) When the Add button is pressed, <u>if the text box is not empty</u>, the text is added to a new item at the bottom of the to-do list.
- (e) When the mouse hovers over an item, it becomes green.
- (f) When a list item is clicked, a line is placed through the item (hint: you can set text-decoration: line-through to achieve this effect).
- (g) When a list item with a line through is clicked, the line through is removed. \bot
- (h) When any list item is double-clicked, it is removed.

An example of what the webpage may look like is below.

My To Do List	
• Zoom • Slack • Teams	
Teams	Add

<html> <head> <meta charset='UTF-8'/> <style> li:hover { color:green; li.done { text-decoration:line-through; (} </style> <script> function strike(){ if(this.className==") this.className="done"; else this.className="; } function remove(){ document.getElementById("todo").removeChild(this); (do covert. ge to lenot By/d(", Tem") /= """) {. function add(){ li.innerHTML=document.getElementById("item").value; li.onclick=strike; li.ondblclick=remove; document.getElementById("todo").appendChild(li); </script> </head> <body> C/ <h I > My To Do List</h I >
 <input type="text" label="New Item" id="item"></input> <button onclick="add();">Add</button> </body> c2 </html>

4. (5 marks)

"The web was designed to be scalable and robust, but it was not designed to be secure."

Discuss this statement with respect to the six characteristics of the Representative State Architecture.

Need 5 good points linked to REST. at least 2 (at least at reest 2 4 REST. secure 4 REST. secure 5 Societies Societies Societies a) HTTP is stateless, requiring cooking in a bronze to maintain state. This compromises privacy and security 6) HTTP is layered. This nears server is pourses may be cached, allowing manin the middle attacky, and you don't know who is serving the regress. .) The cache contains an implicit record of you browsing history. d) Code an demand allows a reaste servers to pourde code that runs on your browser Despite much work by Boonser redos Jour Applets & Flash have been abanded as they cannot ensure the code is at meticion Q) Uniform / kleface means one disurt / Soong an be adapted to consume many series, making He web scalable. lagered achitechne makes neb sodible via g) stateless nems servers on have procies nith out sharing trasactional states making it

golde 2 rob Client server architecture is designed to 60 let the red scale with few server serving clients-1st SEMESTER EXAMINATIONS 10 AGILE WEB DEVELOPMENT 3403/5505 demand de legetes comptation ÇITS3403/5505 Code cannot compol client, but the son Mut code is execut

5. (10 marks) Using your Flask project as an example, describe the process of registering a new user account using a Flask application, noting:

- What HTTP requests and responses will be sent.
- What database operations will be performed.
- How the forms are generated and served to the user.
- How the user-submitted data is validated.

Lei · request register page with HTTi - Form built with WTForms, rendered in to Jinja 3 - HTML abject served as reporse. rostes.p. User fills in fam with un/email and poss hand - HTTP post request start to rec -request recitered at roches. py, -Fern validateron-subarit firms POST (checks (a separte route for POST) - Checks database to confirm UN is anique. - Hash pr, and add use / pr hach to 12,5- Réditet la Login.

11 CITS3403/5505

 $6. \quad (10 \text{ marks})$

a. [5 marks] Consider the flask code below to set up some models linked to a database via SQLAlchemy. Describe the schema of the database as clearly as possible, as well as the models produced.

from app import db from sqlalchemy.orm import relationship _tablename__='students' id = db.Column(db.String(8), primary_key = True) first name = db.Column(db.String(64)) surname = db.Column(db.String(64)) project id = db.Column(db.Integer, db.ForeignKey('projects.project id'),nullable=True) def get partners(self): if not self.project: team = self.project.team team.remove(self) return team class Project(db.Model): _tablename__='projects' project_id = db.Column(db.Integer, primary_key = True) description = db.Column(db.String(64)) team = relationship('Student', backref='project') def __repr__(self): return '[PID:{}, Desc:{}]'.format(\ self.project_id,\ self.description)

2 tables. Id (pk) stdets Stadent (frame Sarane. B | project_id > Proje Project. Many-sli. get-parties. is a pythe faction. acted by SQLA/cheny. 13 a list of studits tem 10

ma In the po Ker giet. tean is n pythen hist 13 1^{st} SEMESTER EXAMINATIONS AGILE WEB DEVELOPMENT 3403/5505 CITS3403/5505

b. [5 marks] Write some unit tests to test the get_partners method in the code from part a. The tests should have with full coverage of the method. Some basic boiler plate code is available below, which you can use if you wish, but it is not required.

```
import unittest, os
from app import app, db
from app.models import Student, Project
class StudentModelCase(unittest.TestCase):
 def setUp(self):
   app.config['SQLALCHEMY_DATABASE_URI']= 'sqlite:///test.db'
    self.app = app.test client()
   db.create_all()
   #initialize test fixtures here...
 def tearDown(self):
   #reset test fixtures here...
   db.session.remove()
   db.drop_all()
 def test get partners(self):
   #tests and assertions go here...
   name ==' main ':
if
 unittest.main(verbosity=2)
```

import unittest, os from app import app, db from app.models import Student, Project, Lab import time

class StudentModelCase(unittest.TestCase):

```
def setUp(self):
  basedir = os.path.abspath(os.path.dirname(___file___))
  app.config['SQLALCHEMY_DATABASE_URI']=\
     'sqlite:///'+os.path.join(basedir,'test.db')
  self.app = app.test client()#creates a virtual test environment
  db.create all()
  s1 = Student(id='0000000',first_name='Test',surname='Case')
  s2 = Student(id='1111111',first_name='Unit',surname='Test')
  db.session.add(s1)
  db.session.add(s2)
  db.session.commit()
  time.sleep(2)
 def tearDown(self):
  db.session.remove()
  db.drop all()
 def test_get_partners_a(self):
  s = Student.query.get('0000000')
  self.assertEquals(s.get_partners(), [])
def test_get_partners_b(self):
```

```
s = Student.query.get('00000000')
s2 = Student.query.get('11111111')
p = Project(description='test')
db.session.add(p)
db.session.flush()
s.project_id = p.project_id
s2.project_id = p.project_id
db.session.commit()
self.assertEqual(s.get_partners(),[s2])
```

if ___name__=='___main___': unittest.main(verbosity=2)

14CITS3403/5505

(vecte mad

nsser

siet

15 CITS3403/5505

End of Paper