Question 1 (20 marks)

a. Name four properties that could reasonably determine the success of a proposed Web application.

b. Name the three fundamental standards that define the World Wide Web.

c. Name two reasons for using cascading style sheets to define HTML document presentation in Web applications.

d. Name three ways of accessing cascading style sheets in an HTML document.

e. When submitting data from an HTML form, when should you use method GET and when should you use method POST?

f. Name the three tiers of the standard Web application architecture.

g. When should input data be validated on the client and when should it be validated on the server?

h. Briefly define a cookie.

i. Name three different methods of maintaining state in Web applications.

j. Name one method of data sanitisation that is performed automatically in Django.

k. HTTPS stands for “secure HTTP”, but how does it work?

l. What is a one-way function? Give one example of its use.

m. If images are stored in a Web application database, name three forms of metadata that must be stored with each image.

n. What does the term ”Ajax” stand for and why are Ajax applications currently popular?

o. Name two key JavaScript features that make Ajax applications possible.

p. Name one framework that simplifies the task of writing Ajax applications.

q. What is the main design pattern used in Django applications? (Don’t just give the abbreviation, use words.)

r. What does the term ”XML” stand for? Give one example of the use of XML.

s. Name the three main components of ”big” Web services.

t. Name four criteria that could reasonably influence the choice of framework to use for Web application development.
Question 2 (20 marks)

a. Every Django project directory should contain at least three important subdirectories. Name them. (2 marks)

b. Every Django app(lication) directory should contain at least three important files. Name them. (2 marks)

c. Every Django model should contain definitions of at least two important methods. Name the two methods and briefly describe why they are important. (4 marks)

d. Name three types of generic views in Django. (2 marks)

e. Give two reasons why template inheritance is important and give an example of the main tag used to implement template inheritance. (2 marks)

f. Suppose class Item defines an item-for-sale model and class ItemForm defines a form model based on class Item. Define a Django view process_form(request) that (1) displays an instance of an ItemForm using template item_form.html if request has method GET, and (2) receives and validates the form data, including image data, to create and save a new instance of class Item in the database if request uses method POST. In case (2), the method should redirect to the URL pattern with name index after saving the new instance. (You may assume all required methods have been imported.) (8 marks)

Question 3 (20 marks)

a. Suppose you are designing a Web application that manages information about movies, directors, actors and so on. Suppose a class Person with fields name, gender and birthdate only is already defined. Consider the following database design for such an application.

```python
from django.db import models

class Movie(models.Model):
    title = models.CharField(max_length=80)
    year = models.IntegerField()
    director_name = models.CharField(max_length=80)
    director_nationality = models.CharField(max_length=80)
    actor = models.ForeignKey(Person)
```

Clearly describe two main reasons why this is a bad database design. (4 marks)

Give a better database design for this application. (6 marks)
b. Consider the following simplified database design for managing student enrolments and grades in courses of a university.

```python
from django.db import models

class Staff(models.Model):
    number = models.IntegerField(unique=True)
    name = models.CharField(max_length=80)
    office = models.CharField(max_length=12)

class Student(models.Model):
    number = models.IntegerField(unique=True)
    name = models.CharField(max_length=80)
    program = models.CharField(max_length=80)

class Course(models.Model):
    code = models.CharField(max_length=7, unique=True)
    title = models.CharField(max_length=80)
    teacher = models.ForeignKey(Staff)

class Enrolment(models.Model):
    course = models.ForeignKey(Course)
    student = models.ForeignKey(Student)
    year = models.IntegerField()
    grade = models.CharField(max_length=2)
```

Clearly describe one nontrivial assumption made by this database design. (2 marks)

Clearly describe one nontrivial restriction that should be captured by the database design for this application but is not captured by this particular design. (Don’t repeat your answer to the previous question.) (2 marks)

Write Django statements using the database API that express the following queries. For example, the statement

```python
courses = Course.objects.filter(year=2009)
```

expresses the query “What courses are offered in (year) 2009?”.

(i) What is the title of the course whose code is “3515ICT”?  
(ii) What are the names of the staff members who teach courses titled “Information Systems”?  
(iii) What are the names of the students who received a grade of “HD” in the course coded “4001ICT” in year 2008?

(6 marks)
Describe how you would design a Web application in Django to provide a new social networking site for photographers, such as Flickr or Picasa Web Albums. The new site allows photographers to publish and organise their photos, view other photographers’ photos, and comment on other photographers’ photos. Such a system should provide the following capabilities:

Users may register to become members, and members may login and logout. Each member has a unique username, first and last names, an email address, and other personal information (which may be ignored for this question).

Logged-in members may post (i.e., publish or upload) photos to their photostream (the list of all photos they have posted). Each photo has a member-defined name, an image, a thumbnail (a small system-generated version of the image), a poster (the member who posted the photo), the date it was posted, and other metadata (which may be ignored for this question.)

Logged-in members may create (initially empty) sets of related photos. Each set belonging to a particular member must have a unique name (e.g., People, Landscapes, Birds), a brief description, and a set of photos posted by that member. Each photo posted by a member may belong to several different sets created by that member, or to no sets at all.

The site’s overall home page displays the list of all members’ names, and other information (which may be ignored for this question). (Note that this requirement is different from the actual home pages of Flickr and Picasa Web Albums.)

Users may select a particular member’s name from the list on the overall home page, which leads to that member’s home page. Each member’s home page contains that member’s name, the member’s photostream (as a list of the thumbnails of the member’s photos, most recently posted photos first), and a list of that member’s sets, if any.

Selecting a set from a member’s home page leads to a set page that contains the member’s name, the set’s name and a list of the thumbnails of the photos in that set.

Selecting a thumbnail from a member’s photostream or from a member’s set page leads to a photo page that contains the photo’s actual image, any comments on the photo (see below), and other information about the photo.

When a member logs in, they are taken to their own home page.

Logged-in members who are viewing one of their own photos may add the photo to a set or may delete the photo (from their photostream and from every set it belongs to).

Logged-in members may publish a comment on any photo they are currently viewing. Any user viewing a photo will see all comments made on the photo, after the photo’s image, most recently published comments last.

Your description should include the following aspects of the implementation:

a. A database design (models.py): Describe all database models required, indicating the purpose of each model and field by carefully chosen names or by comments, and including any constraints that must hold. (8 marks)

b. A set of URL patterns (urls.py): For each URL pattern, describe what (Django) view it is associated with. (Views may be user-defined views, generic views, or other Django library views.) (6 marks)
c. A set of Django views (views.py): For each user-defined view, describe carefully, in English, what it does, including whether it renders a template or redirects to some URL. (6 marks)

d. A transition diagram: Give a transition diagram, naming each node by a URL pattern, indicating whether each node generates HTML output or is just executed for its effect, and including all transitions between nodes in the diagram. (4 marks)

Note that you do not have to define models to represent members or comments; you can use the Django authentication framework django.contrib.auth and the Django comments framework django.contrib.comments.

Question 5 (16 marks)

a. Briefly describe the main roles of XML and XPath in Web application development. (4 marks)

b. Briefly state what a news feed format is and what a news aggregator application is.
   Give two examples of news feed formats and two examples of news aggregator applications. (4 marks)

c. Briefly describe the main concepts behind RESTful Web Services.
   Give two examples of existing RESTful Web Services. (4 marks)

d. Briefly describe the purpose of the tools Struts 2, Hibernate, Ant and IntelliJ in Web Development using Java EE, relating these tools to Django equivalents where possible. (4 marks)