# An Online Newspaper App

**Group**: 2-3 students
**Weight of final mark**: 35%
**Deadline**: Friday 4 December

THE TIMES theguardian THE Sun

**Description**: Your task is to develop an **online newspaper web app** using the Django framework. The website should provide the following functionalities (*you must implement the features in this order, e.g. you should not implement an ADVANCED feature if you have not first completed all BASIC and INTERMEDIATE features*):

| | |
|---|---|
| **BASIC** | **B1.** Users can create an account on the site and and able to login into their account. The user's account should contain at least an email address and their date of birth. |
| | **B2.** Users can browser news articles, and news article are grouped into categories (e.g. Sport, World, Finance,…) |
| | **B3.** Users are able to select (in their profile page) their favourite news categories, and should only be shown articles within their favourite categories. |
| | **B4.** Users should be able to "like" a news article by pressing a "like button". |
| **INTERMEDIATE** | **I1.** User's profile also contains a profile picture. Users are able to add/update or delete their profile picture. |
| | **I2.** Users are able to post comments on an article, and are able to reply to previous comments on the same article. They should also be able to edit and delete their own comments. |
| **ADVANCED** | **A1.** Liking an article and posting/editing/deleting a comment are done via Ajax. |
| | **A2.** Users receive a welcome email upon signing up to the web app. |
| | **A3.** Automated testing is included (using selenium) testing basic features B1 (account creation and login), B4 ("liking" an article), and I2 (posting a comment and subsequently deleting that comment). |

**Emails**: When implemented feature A2 (welcome email), please create a temporary Gmail account, and use that to send emails, as you will need to submit the code for your app which might contain the password for the email account (unless you are using environment variables on the server to store your passwords).

**Outcome**: Once fully tested, your application should be deployed to the school's OpenShift web servers (to be discussed in week 7) — one deployed app per team. The application should be ready for testing, with at least 10 different news articles, 3 different news categories, and at least 3 test users. The single zip file you submit on QM+ must include:

- the **code** for your app (including the **requirements.txt** file listing the dependencies for your project),
- a **README** file containing your group number, name and email of members, the URL of your deployed application, the username/password for the admin user, and the username/password for the 3 test users.
  **You must also describe in the README file the contribution of each team member to the project (one/two sentences per group member).**

MARKING SCHEME ON NEXT PAGE

### Marking scheme

| Maximal mark<br>(100 marks) | Achievement | |
|---|---|---|
| **Deployment**<br>**(10 marks)** | App successfully deployed | **10** |
| | Deployment attempted but not fully working | 3 - 9 |
| | Deployment attempted but not working at all | 2 |
| | Deployment not attempted | 0 |
| **Functionality**<br>**(60 marks)** | All **BASIC** features working as expected | **30** |
| | All **BASIC** and **INTERMEDIATE** features working | **45** |
| | All **BASIC**, **INTERMEDIATE** and **ADVANCED** features working | **60** |
| | Features partially working | 5 - 55 |
| **Data modelling**<br>**(10 marks)** | Excellent use of Django models, e.g. using Django's User model, class methods and correct relationship fields | **10** |
| | Incorrectly modelled data or no use of Django's model features | 4 - 9 |
| **Frontend**<br>**(10 marks)** | Excellent use of Bootstrap, jQuery and ajax requests | **10** |
| | Partial or no use of Bootstrap, jQuery or ajax | 0 - 9 |
| **Views + Templates**<br>**(10 marks)** | Code duplication avoided using decorators and template hierarchy. Good use of URL reversing. | **10** |
| | Some duplication of code and/or hardcode URLs | 0 - 9 |
| **Total** | | **100** |