Assessment of Application Programming 2019-2020

UNIVERSITY OF PORTSMOUTH

Unit Coordinator	Dr Matt Dennis <matt.dennis@port.ac.uk>
Issued	October 2019
Code	U30221 Application Programming
Feedback by	20 working days after submission

Submission Summary

Topic	**Available from**	**Deadline**	**Weight**	**Format**
In-Class Test	2019-10-21 14:00	2019-10-21 16:00 (2 hours)*	25%	JS
Programming Assignment	2019-12-09	2020-01-13 12:00 (noon) GMT	25%	JS
Web Application	Now (below)	2020-03-27 12:00 (noon) BST	50%	ZIP

Notes and Advice

- The In-class test is an exam-type assessment, so if you are not present you will receive zero. If you are unwell (usually a valid extenuating circumstance) you will be able to take the test in the referral/deferral period at the end of the year.
- The late deadline for the Server Unit tests is ten *working* days after the deadline (so, 2020-01-27).
- The late deadline for the Web Application is ten *working* days after the deadline (so, 2020-04-14).
- If you do not have valid extenuating circumstances late work will be capped at 40%.
- The Extenuating Circumstances procedure is there to support you if you have had any circumstances (problems) that have been serious or significant enough to prevent you from attending, completing or submitting an assessment on time.
- ASDAC are available to any students who disclose a disability or require additional support for their academic studies with a good set of resources on the ASDAC moodle site
- The University takes plagiarism seriously. Please ensure you adhere to the plagiarism guidelines.
- Any material included in your coursework should be fully cited and referenced in APA format (sixth edition). Detailed advice on referencing is available from http://referencing.port.ac.uk/
- Any material submitted that does not meet format or submission guidelines, or falls outside of the submission deadline could be subject to a cap on your overall result or disqualification entirely.
- * Students granted extra time by ASDAC will get that after the normal time.

If you need additional assistance, you can ask your personal tutor, learning support ana.baker@port.ac.uk and xia.han@port.ac.uk or your lecturers.

Web Application Brief

You are required to design a Dog Breeding App where owners of dogs can search for a suitable mate for their pets. Much of the design and layout of the app is up to you. One could envisage a list, grid, or swipe left/right user interface.

Application Requirements

Core capabilities:

- Dog owners can create a profile for their dog as available for breeding.
- Dog owners can see profiles of potential mates for their dogs.
- Dog owners can arrange to meet.
- Owners of male dogs should see bitches, and vice versa.

Some additional features we can imagine are:

- Constrain results by breed, distance, age or Kennel Club registration status.
- Multiple profile listings for owners, so both sexes are visible.
- Profiles could have one or multiple photos attached.
- Verification notes left on profiles by other users.

These additional features are only *suggestions* – you are free to invent & implement other capabilities.

Assessment Requirements

You will build an application from scratch, fulfilling these requirements:

1. You are to specify and construct this tool using a combination of HTML, CSS, and JavaScript.
2. **IF** a database is required, you may only use services available on your VM (such as SQLite, Postgres and MySQL).
3. The application must start, and be reachable, via HTTP on port 8080 when run on our test machine.
4. The application must launch with `npm start`
   - If your application requires a configuration step, this must be achieved via the command `npm run setup` and documented in the README file – we will not undertake any other manual steps.
5. If the server does not start, we will assess the work based on the source code without the benefit of seeing it run.
6. You must include a `README.md` file that explains key features, how to use them, details your design and implementation rationale, and lists unfinished and future work.

Notes

- This is individual work.
- If you need help you should…
    - Use the classes to ask your tutor.
    - Discuss the problem with your peers (it's ok to discuss, just **don't work together on a shared solution**).
- Your task is to show (through your work) the extent to which you have met the learning outcomes for the module, which are:
    - Design, implement and test algorithms to solve problems using appropriate data types and control structures.
    - Design, implement and test programs based on a supplied specification.
    - Describe and analyse fundamental programming concepts and techniques.
    - Recognise and select appropriate techniques that might be suitable for managing coursework or small commercial projects.
    - Employ techniques to identify and implement the information requirements of simple data and documents (for example, define and create a website on a given topic and identify how form and content should be separated therein).

Marking Scheme

The Dog Breeding App will be graded using ***Academic Judgement***, using this guide:

Topic	Description	Weighting (%)
Functionality	How appropriate is the design (data, code, architecture)? Does it all work? How much does it do? How much is your own work as opposed to libraries?	30
Maintainability	Code style, comprehensibility and maintainability. This includes formatting, file structure, naming - everything that can help your work live on and be useful *after* it is graded, including how well the code and any documentation communicates any concepts necessary to understand the architecture and configuration of the system.	30
Usability	Ease-of-use of your system, including the use of event-driven input, intuitive UI design, etc.	10
Delivery	Does it install and start as required?	10
Invention	We will award up to 20% bonus marks for unusual qualities, strengths, creativity and invention not otherwise prescribed here.	20