Project 2016

Emerging Technologies

Due: 01/12/2016

The following document contains the instructions for Project 2016 for Emerging Technologies. The project will be worth 100% of your mark for this module. However, it is split into three parts: a programming component worth 40%, an architecture component, also worth 40%, and finally a presentation component worth 20%. The project will involve teamwork and will be managed through the use of GitHub.

You are required to develop a single-page web application (SPA) [7] written in the programming language Go [6]. You must devise an idea for a web application, decide on an architecture, create a development plan, write the software, write documentation explaining how the application works, and write a short user guide for it. You will also have to give a presentation of your project at the end of term.

Teamwork

For this project, you will work as part of a team. Teams will be selected by the lecturer at the beginning of the module. Effort will be made to ensure a balance of skills on each team.

Teamwork experience is frequently requested by employers. This is usually coupled with the ability the use of collaborative software such as GitHub. Furthermore, should you plan to create your own startup, working alone is not considered a great idea, as evidenced by Y Combinator co-founder Paul Graham's essay on mistakes startups make [5].

GitHub

GitHub must be used to manage the development of the software, including the use of GitHub Issues [2]. GitHub issues allow collaborators to track project progress through bug reports, milestones and labels.

Teams will have the opportunity during the weekly timetabled computer labs to work together on the project, and to ask the lecturer for advice. Your team's GitHub repository will be reviewed each week in these labs. The lecturer will add issues and comments to the team's repository to record these interactions. It is likely that these interactions will heavily affect the marking of the project.

Dealing with problems

There are a few common problems that arise in teamwork [8]. One you may encounter is a difficulty in getting started. To counteract this, ensure your team members have been properly introduced to each other, and everyone has explained how they think that they can best contribute to the project.

Another common problem is a member of the team not contributing. Before coming to this conclusion, members of the team should ensure that the issue is not justa symptom of ineffective communication. Some members of the team will naturally be less talkative than others, and this may not reflect their work effort. Likewise, some members of the team will naturally be more forthright. It is important that all members of the team feel that they have opportunities to be heard.

However, should a team decide to, they may dissolve their team, fork the project, and pursue the forked projects individually and/or in smaller groups. This might also be appropriate if some team members be consistently absent, non-responsive and are having a negative effect on the project. This means that from that point forward, part of the team will work on a separate codebase. GitHub facilitates this process, as it is fairly common in open-source projects, with examples abound on GitHub [1].

Technology

You can use any combination of the following technologies in your project. However, Go must be the predominant language used, especially for the API. Any other technologies, including libraries and frameworks, you wish to use must be agreed to by the lecturer.

Languages — Go, HTML, JavaScript, CSS, Typescript, Less, Sass

Libraries — Anulgar.js, jQuery, React, Bootstrap, Skeleton, Ember.js.

Frameworks — Macaron, Beego, Martini, Gorilla, GoCraft, Revel, Web.go.

Databases — Any SQL RDBMS, CouchDB, MongoDB, Neo4j, Redis.

Submission

Your team's GitHub repository, along with the Issues tracker and all other GitHub facilities, will form the main submission of the project. You must also present your project at the end of the semester. The exact dates of presentations will be determined at a later date, in consultation with the class. One team member can do the whole presentation, but everyone must be present to answer questions. Presentations will be limited to 10 minutes.

Quality

A good project submission will demonstrate a team that worked well together to create a useful and easy-to-use web application. The code base will be well thought-out and contain extensive comments. Their will be useful help documents and README files. The issue tracker will show that the project was well managed. Remember that programming is only part of the task.

Student Conduct

You should familiarise yourself with GMIT's code of student conduct [3] and also the policy on plagiarism [4]. In particular, note two things. First, students are expected to treat other students and staff politely and with courtesy. Second, it is assumed that all work you submit is being presented as your own work, unless referenced otherwise.

Marking scheme

The following marking scheme will be used to mark the project. Student should note, however, that in certain circumstances the examiners overall impression of the project may influence marks in each individual component.

40%	Programming	Well-written code; logical directory structure; extensive commenting; extensive git commits.
40%	Architecture	Good separation of concerns; clear API, use of new technologies.
20%	Presentation	Well-prepared; interesting parts of project highlighted; confident.

Expected standard

Please note that this is a level 8 module. You should be aware that the standard required for submissions at level 8 (fourth year) is higher than at level 7 (third year), which in turn is higher than at level 6 (first and second year). Significant effort is made to ensure that the standard is fair and consistent across third level institutes, both nationally and internationally. The standard we set for modules in computing is informed by Quality and Qualifications Ireland's Award Standard for Computing [9]. Below is a particularly relevant selection of the learning outcomes contained in that document.

Level 8 (Year 4)

The learner will be able to:

- *describe the limitations of some current computing theories.*
- *evaluate information through online research.*
- *model and design complex computer-based systems in a way that demonstrates comprehension of the trade-off involved in design choices.*
- *demonstrate mastery of a complex and specialised area of skills and tools;*
- *manage one's own learning and development, including time management and organisational skills.*
- *manage a computer-based project throughout all stages of the lifecycle.*
- *apply quality concepts to products and processes of own work.*

Level 7 (Year 3)

The learner will be able to:

- *integrate concepts learned across a variety of subject areas.*
- *identify relevant material on a given topic from available information sources.*
- *succinctly present rational and reasoned arguments to a range of audiences.*
- *develop innovative solutions to pragmatic situations.*

- *recognise the suitability of a given solution to a problem.*
- *apply knowledge learned in new situations.*

Level 6 (Years 1 and 2)

The learner will be able to:

- *describe, recognise and apply best practices in computing.*
- *apply knowledge in a practical setting under supervision.*
- *demonstrate the capacity to learn new knowledge and skills.*
- *use troubleshooting strategies and techniques in correcting a variety of computer hardware and software problems.*
- *implement computer based systems solutions to well-defined problems.*

References

[1] GitHub. Fork a repo.
`https://help.github.com/articles/fork-a-repo/`.

[2] GitHub. Mastering issues.
`https://guides.github.com/features/issues/`.

[3] GMIT. Code of student conduct.
`http://www.gmit.ie/sites/default/files/public/about/docs/code-student-conduct-2016-17.pdf`.

[4] GMIT. Policy on plagiarism.
`http://www.gmit.ie/sites/default/files/public/general/docs/10-no2-plagiariam-3.5.pdf`.

[5] Paul Graham. The 18 mistakes that kill startups.
`http://paulgraham.com/startupmistakes.html`.

[6] Google Inc. The go programming language.
`https://golang.org/`.

[7] Google Inc. Writing web applications.
`https://golang.org/doc/articles/wiki/`.

[8] The University of Queensland Student Services. Problems associated with group work.

[9] Quality and Qualifications Ireland. Award standards – computing. http://www.qqi.ie/Publications/Computing%20-%20QQI%20Awards%20Standards.pdf.