

BSc. Level 8 courses  
Module: Web App Development 2

Assignment 1 – React SPA  
(Client-side)

Due date: 08/11 /2019

# Assignment 1

- Develop a SPA, using React, for a context of your choice:
  - Productivity, Social networking, E-Commerce, etc
  - Alternative: Extend one of lab apps (Lower weighting)
- Prototype standard.
- Constraint: Must use create-react-app and Storybook
- CA percentage: 60%.
- Git log is mandatory
- No robust persistence/server-side necessary.
- Deliverables:
  - GitHub repository, including README (details later; mandatory).

# Grading spectrum

- **Good (40-60%) – Theme: Foundation-standard app.**
  - Features/Qualities:
    - Demonstrates all the React foundation topics covered in the module.
    - Basic routing.
    - Interactive and dynamic behaviour.
    - Part-CRUD functionality.
    - Data model: Stub API; 2 entities
    - Storybook support.
  - Alternative: Adapt one lab app to a different context, plus new features/view.

# Grading spectrum

- **Excellent (60-80) – Theme: Feature-rich, demo App.**
  - **Features/Qualities:**
    - **App has a coherent and (almost) complete feature-set. Suitable for demonstrating the Single Page App model.**
    - **Parameterized URLs.**
    - **Header/Footer navigation; Extensive data Hyperlinking.**
    - **Full CRUD functionality.**
    - **Mock Backend integration (json-server)**
    - **Data Model – 2+ entities; 1+ Relationship.**
- **App is YOUR idea. Not based on labs (OR found on GitHub)**

# Grading spectrum

- **Outstanding (80-100) – Theme: Independent learning.**
  - **3rd party authentication (e.g. Firebase); public/private views.**
  - **Advanced React (e.g. Context API),**
  - **Storybook addons,**
  - **3<sup>rd</sup> party Web API; 3<sup>rd</sup> party components.**
  - **Deployed (e.g. Surge),**
  - **Other.**

# ***Git is your assignment DIARY.***

- **Git commits:**
  - Should tell a clear story of your progression through the assignment.
  - Methodical, distributed workflow. .
    - Don't fake it!!
  - Some 'Bug Fix' messages are okay,
  - Commit regularly.
  - Rollbacks should work (Don't commit failing tests)
- **Use .gitignore correctly.**
  - Especially at git init stage.
- **PLAN**

# GitHub README

- Mandatory.
- Determines your final mark.
- Follow the structure provided.