Paired Assignment 1

Due Friday by 11:59pm **Points** 15

Assignment 1:First Step in Building Modern Software

Paired assignment (15%)

Overview

As discussed during the lectures, there are many technologies and tools you can use to build a modern software product. In this assignment, you are asked to familiarize yourself with some of the key tools you can use to build two of the most common types of software (web and mobile). You can see this assignment as a preparation for your project or the first step in familiarizing yourself with the modern software engineering world. You will work in **teams of two** to build a tech stack, for a mobile or web application. The application you will be building is a **simple checkout price calculator** with basic functions. You can think of the functionalities as that of the system that is used during checkout. Use cases can be adding items, removing items, applying discounts, adding tax, and any other realistic functionality you want to add **but keep it simple.** (you don't need any additional functionality, like save, load, advanced UI, etc.)

As you can see, the focus of this assignment is **not on the complexity of the software but the infrastructure** you need to establish to be able to develop the software.

Assignment breakdown

This assignment has two major paths:

1. Web application: “How do we build a web application that can easily be developed, tested, maintained, and deployed?” is the question leading this part of the assignment. You will research various solutions and steps needed and build on top of what we have talked about in the lectures. You can consider this web application would need (not part of the assignment) some standard functionality such as login, search, post comments, etc. and no special features (e.g., advanced visualizations, AI components, etc.). Your application must be [responsive (https://en.wikipedia.org/wiki/Responsive_web_design)](https://en.wikipedia.org/wiki/Responsive_web_design).
2. Mobile application: “How do we build a mobile application that can easily be developed, tested, maintained, and distributed?” is the question leading this path. You can make an assumption about whether your application needs to run on Android, iOS, or both. But you must clarify that assumption in your report.

in your report.

The process you follow for each path is very similar. However, the end result will be different. Please choose the path that you think your project will be in.

Next Steps

Your customer has chosen you as their trusted advisor and asked you to recommend a solution for their needs. You can start your research from the roadmaps we have seen during the lecture **(https://roadmap.sh/)** . Consider three main technologies to use for each part. For example, for the backend of your web application, you can look at Node, Python, or Go, or any other language you find interesting. For the frontend of your application, you can look at Vue, React, Angular, or any other language/framework. You will then need to compare the options you have chosen. Considerations can include (but should not be limited to):

1. Ease of development for you (e.g., you know Python but Go may be new to you)
2. Maturity of the language/technology and the libraries available for integrations/UI
3. The domains covered by the technology/language (e.g., Python also supports ML projects while Javascript can do both frontend and backend).
4. Popularity of the language/technology. You can look at HackerRank 2020 Report **(https://research.hackerrank.com/developer-skills/2020)** , Stackoverflow 2019 Survey **(https://insights.stackoverflow.com/survey/2019)** , or JetBrains Developer Ecosystem report **(https://www.jetbrains.com/lp/devecosystem-2020/)** .
5. Performance, scale and speed of the solutions built with these technologies.
6. Any other criteria you think is important and needs to be considered

Your tech stack needs to include solutions for the following:

1. Frontend technology
2. Backend (web or mobile)
    1. Needs to include Testing Infrastructure (e.g., PyUnit for Python unit testing). You don’t need to have comprehensive tests. One test is enough.
3. Continuous Integration & Continuous Delivery (CI/CD)
4. Database (we will talk about databases in one of future lectures in more depth)

Your setup needs to support:

1. One development environment and a production environment.
2. Automated deployment on merge at least to the development environment.

1. You can use any of the CI/CD tools.
2. Your deployment needs to include running test(s)

Deployment

You are expected to make your applications available for your TA to see, test and grade (mobile and web). You will need to “deploy” your applications. We will use the word deploy loosely here meaning that you need to have your web application available via a url and find a way to share your mobile app with the TA (e.g., apks, expo, developer account sharing, video recording, etc).

Expected Submission

You will submit three things in your GitHub repo.

1. A report containing the options you considered for your path (mobile and web). This report needs to include at least three options and a comparison of pros and cons of each of them. In the comparison, you can consider the following elements:
2. Your final solution that you decided to choose and the key reasons for that decision.
3. Please provide a summary of your analysis (Max 1000 words. This doesn’t mean you have to write 1000 words. Quality is more important than quantity).
4. Application with all the details and instructions needed for your TA to see and test your application. Please note that if your instructions are not clear enough to be easily tested, you will lose 5%.

How to Submit

You will be submitting your assignment on Github. The following instructions will allow you to form a pair and create a repository for submission:

1. You must form an assignment pair on Quercus by navigating to the ‘*People*’ tab, then ‘*Assignment 1 Pairs*’. Add yourselves to a pair (e.g., **Assignment 1 Pair 1**). The number that you choose will be your pair number for this assignment
2. You must either create a repository using the following link **(https://classroom.github.com/g/AY6_gAid)** or join an existing repository that your partner has created. You will use this repository to submit all material related to your assignment. Note that *‘assignment1-pair’* is automatically included in your Github team name. In other words, only enter ***<pair-number>-<member1-github-id>-<member2-github-id>.***

Marking Scheme

Report (40%): Your report needs to contain the tools and languages you researched and provide a detailed comparison of them leading to your final choice.

Deployed application (40%): You need to have deployed your application and setup CI for your

Deployed application (40%): You need to have deployed your application and setup CI for your applications

Easily available and testable application (20%)

Evaluation

Your assignment will be evaluated according to the following rubric

Application	Component	Mark	Weight	Notes
Web/Mobile Application	Report	__ /400	40%	100 points per comparison
	Deployed Application	__ /400	40%	100 points per criteria
	Testing Availability	__ /200	20%	100 points per criteria

We will compute your mark by assigning each question a percentage according to the standards set below and take the weighted average. Each point is really a percentage point (i.e. 85 points on a question is 85% with regards to the standard below.

Evaluation Criteria for Report (Web / Mobile)

You will be comparing three main technologies for each of your four web application components or each of your four mobile application components. Each of you comparison will be evaluated according to the evaluation criteria outlined below:

Grade	Grade Expectations
100% *(Outstanding)*	• above expectations in quality or creativity of work (e.g. critical thinking, creative approach, etc.) when performing comparisons between technologies • More than five considerations included in answer • meets expectations for 80%
80% *(Good)*	• answer meets all expectations and addresses all five considerations listed in the assignment • answer has both content and justification (i.e. what/how and why)

(Good)	- answer has both content and justification (i.e. what/how and why)
70% (OK)	- some expectations were not fully met within the question either because the answer is unclear or no clear justification was provided for the comparison (e.g. you mention what but not why) - one major (or a relatively small number of minor) point(s) were not addressed in the question (i.e you did not talk about the domains covered by the technologies)
60% (Below Expectations)	- more than one major consideration was missed or not addressed within the answer to the question - Multiple minor points are lacking clarity or clear justification
50% (Minimal)	- The provided answer does not cover any of the required details or is too high-level (i.e. an answer was provided but met no expectations or was not justified)
0% (Missing or extremely low quality)	- multiple (or all) points missed or very little content provided for the comparison - No comparison is performed

Evaluation Criteria for Deployed Application (Mobile & Web)

Grade expectations

Criteria	**Outstanding - 100%**	**Great- 80%**	**70% - Good**	**60% - Below Expectations**	**50% - Minimal**	**0% - Mis**
1 - The application is functional	Meets expectation for 80% Complex/additional functionalities are included in the application	All standard functionalities are present (i.e the application meets requirements)	Missing one or two standard functionalities Moving between features (e.g.	Missing three or four standard functionalities Little to no cohesion	Missing majority of standard functionalities No cohesion between	

		The workflows (e.g. going from feature or view to the next) can be executed	features (e.g. workflow) isn’t fully functional but one can access each feature independently	cohesion between features or application requires resetting between features	between features - it is impossible to move from one feature to another	NO SOFTW PRODU EXISTS CRITER COMPL MISSIN
2 - The application’s UI is present and usable	Meets expectation for 80% UI is entirely free of visual defects UI is visually/aesthetically pleasing UI is highly usable and incorporates design principles	A UI was built UI is mostly cohesive and clear	A UI was built for most features UI requires some testing or constant reference to documentation to use or understand	Very few features contain a UI UI is very non-intuitive and without documentation testing would be difficult	No UI was complete but some UI code exists	
3 - Testing infrastructure implemented	Meets expectations for 80% Testing infrastructure is comprehensive (beyond just unit tests)/ comprehensive/fully and easily extensible going beyond unit tests	Testing infrastructure set up with one test case that can be run successfully Testing infrastructure can be adapted to develop further testing	Test infrastructure is fully set up but is not functional (i.e. the test does not run successfully) Testing infrastructure cannot be adapted (or must be modified	Test infrastructure is only partially completed	No testing infrastructure but some code for testing exists	

		further testing when expanding application functionality	modified extensively) to develop more tests			
4 - Continuous Integration and Continuous Development setup	Deployment goes all the way to production with every push to master running all infrastructure successfully with high-quality tests	CI/CD is setup and runs everything successfully with every push	CI/CD is set up	CI/CD is fully set up but is not functional	Team has evidence of unsuccessful but attempted CI/CD setup	

Evaluation Criteria for Testing Availability (Web & Mobile)

	Grade expectations					
Criteria	**Outstanding - 100%**	**Great- 80%**	**70% - Good**	**60% - Below Expectations**	**50% - Minimal**	**0% - Missing**
1 - The application is deployed	Meets expectations for 80% Product is deployed fully through automation	Product is available to a tester without setting it up on a developer machine Product deployment is easily repeatable	Product deployment was completed manually with interventions at most or all stages of deployment (e.g adjusting multiple parameters, manual branch adjustments, directly accessing deployment environment)	Product is deployed but cannot be accessed and requires team to present their work	Team has evidence of attempted but unsuccessful deployment	NO SOFTWARE PRODUCT EXISTS OR A CRITERIA IS COMPLETELY MISSING

2 - Instructions for testing are provided	Meets expectations for 80% Above expectations in quality or creativity of instructions (e.g., highly detailed, video demonstrations, GIFs, etc.)	Clear instructions are provided on how to use the application from the tester's perspective Clear steps provided for using each feature that has been built Instructions can be executed as provided	Instructions are provided for using the application, but some assumptions had to be made, or some ambiguity in instructions present Some of the instructions cannot be executed as provided	Instructions are very unclear or ambiguous, and requires the tester to make many assumptions to use the application Some missing steps	Provided instructions are insufficient for using the application from the tester's perspective Steps not provided for using each feature that has been built, or many missing steps None of the instructions can be executed as provided