

**FULLSTACK ASSESSMENT MATERIAL RELEASE**

(Alternative) THEORY QUESTIONS

|  |  |  |
| --- | --- | --- |
| **SECTION TYPE** | **TOTAL MARKS AVAILABLE** | **NOTES** |
| **Design heuristics** | 10 |  |
| **Agile** | 10 |  |
| **React** | 20 |  |
| **40 marks available total** | | |

**Important notes:**

* This document shares the first section of the Full Stack Exam which is composed of 3 Full Stack Theory Questions
* You have 24 hours before the exam to prepare.
* If any plagiarism is found in how you choose to answer a question you will receive a 0 and the instance will be recorded. Consequences will occur if this is a repeated offence. You can remind yourself of the plagiarism policy [here](https://drive.google.com/file/d/1k9UaGOR7hx54QRZ8jvp2jtC4P-8_Rs4F/view?usp=sharing).
* Answers need to be explained clearly and illustrated with relevant examples where necessary. Your examples can include code snippets, diagrams or any other evidence-based representation of your answer.

*Questions begin on the next page*

|  |  |
| --- | --- |
| **Theory Questions** | **40 total** |

1. Explain why it’s important to be mindful of consistency and standards when designing an app. *Make sure to point out the advantages and if any the disadvantages.* (10 marks)

The rules of consistency and standards are important as they mean a new user interacting with your app will not become confused. This applies to both the UI and behaviour of a given product.

There are several advantages to following these principles:

**Predictability:** Most users do not want to learn how to use a new app each time. By applying these principles, it is possible to create a predictable app that allows users to apply their prior knowledge.

**User familiarity:** Consistency is important for creating user familiarity. Consistency between apps means users do not need to learn new functionality every time. Consistency within apps means users do not need to learn new representations of tasks every time.

**Efficiency in development:** If an app is based off existing standards, tools and libraries it can increase efficiency and speed up development.

**Brand identity:** Following these standards can create an easily recognised brand with associated visual elements.

There are some disadvantages to following these principles:

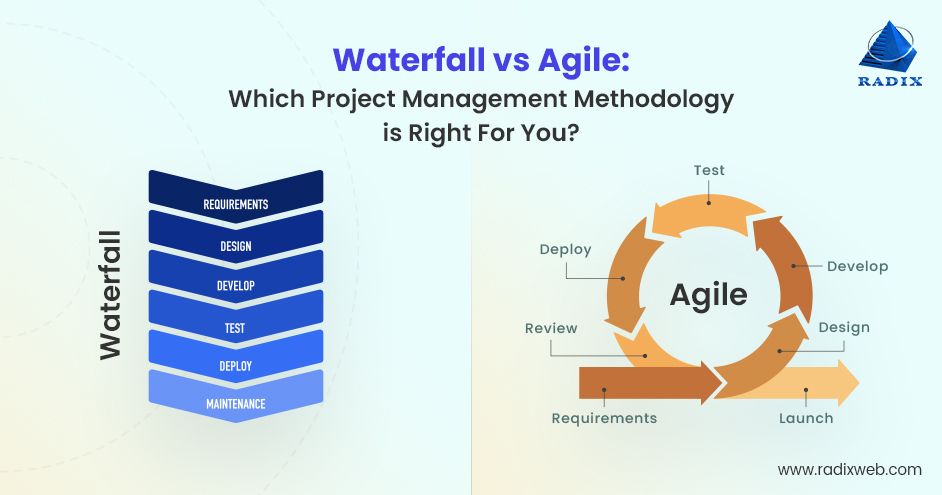
**Limited creativity:** There is the risk of stifling creativity if too much emphasis is placed on following what has come before. Innovative features may be difficult to integrate.

**User diversity:** Whilst a standard design may be suitable for one user group, user diversity means the app may not appropriately cater for all users. For example, following design templates for users who read from left to right will not appropriately target those who read from right to left.

1. What is Agile testing? What testing is done during Agile? What is the difference between a traditional Waterfall testing and Agile testing? (10 marks)

Agile is a popular software project management framework. It is based on the principle of continuous cycles of development and feedback. Agile testing is a testing methodology that follows the same principles as agile development. This means testing is started early in the software development process and is continued throughout. To follow this method, development teams can either ensure they are releasing their code for testing regularly, i.e. up to multiple time a day, or can adopted a test-driven approach where the tests are written prior to any product development to define the desired functionality. In both cases tests should be performed alongside software development.

Waterfall is an alternative project management framework that focuses on a more linear model. Software is development sequentially, with each phase of the development process completed prior to moving to the next phase. The main difference between waterfall testing and agile testing is that in waterfall, testing is a separate phase that does not occur alongside development. Instead, testing is carried out only after development has completed.



* 1. What are the limitations of React? (10 marks)

**Steep learning curve:** There is a steep learning curve to getting started with React which requires prior knowledge of JavaScript. There are complex React concepts that users need to understand, including virtual DOM, JSX syntax and state management.

**Boilerplate code:** Another disadvantage is the extensive boilerplate code that is required to run a ReactJS application. This means they are often larger and more complex.

**Limited functionality:** React is a view library not a framework. This means a developer must rely on additional libraries to create a full application. Whilst this does give flexibility when choosing which external library is appropriate for your

app, it can add complexity and increase development time properly integrated these libraries.

**Fast development:** There are constantly new tools and dependencies being added to React. Whilst this fast development can be an advantage, often the documentation becomes outdated, and it becomes difficult to keep up with the new tools and features.

**Performance issues:** Certain features, for example, components that are not designed properly, a state management system that has grown too large, and design that causes unnecessary re-renders, can all cause performance issues.

* 1. What are Error boundaries in React and How do you handle error handling in a React application (10 marks)

Error boundaries are React components that can catch JavaScript errors anywhere in the component tree and below it. This capability handles errors and therefore prevents the whole application from crashing. Instead, the error is logged, and an alternative UI displayed.

To handle errors in React applications you can import and use the react-error-boundary library. You can then use the error boundary component to wrap around a component to catch any errors that arise from this piece of code. For example:

A screenshot of a computer program

Description automatically generated

The error boundary component has a fall-back render prop which accepts a function or React element to display if an error is caught. It also contains a prop that can reset the state of the component. Finally, the error boundary library provides a Hook that can be used to throw errors within the function components. This is particularly useful for errors that may not be caught by the error boundary. Overall, using this library makes the process of catching and handling errors streamlined and easier.