CODE FIRST GIRLS

CFGDEGREE

FULLSTACK ASSESSMENT MATERIAL RELEASE

THEORY QUESTIONS

SECTION TYPE	TOTAL MARKS AVAILABLE	NOTES
Design heuristics	10	
Redux	10	
React	10	
Agile	10	
40 marks available total		

Important notes:

- This document shares the first section of the FullStack Assessment which is composed of 4 FullStack Theory Questions
- You have 24 hours before the assessment 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.
- 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. **In design Heuristics, what does the term "advantages of Matching between system and the real world" mean? What are the advantages?**

Answer Design heuristics is like creating a design that reminds you of something you see or do every day. This makes it easier to understand. An example would be like the battery symbol on a MacBook. When the battery icon is full and bright it means the battery is full. When the icon is less bright and not full it means the battery is low. This design makes it easy for understanding because it acts like a real-life battery.The advantages of this approach are:

- Easy to Understand: User don't need to learn new stuff just use what User already know from everyday life

- Universal: It doesn't matter who you are or where you're from these symbols or designs are based on experiences we all have, so everyone gets it

- Fewer Mistakes: The symbol works like you expect it to so Users are less likely to make mistakes when using it

- User Expectations: User have a natural understanding of how things should work based on your real-life experiences. This design matches those expectations, so they will feel comfortable and in control

So it's like making a design user-friendly is ensuring that it acts like things in the real world. This way users don't have to think too hard on how to use it - it just makes sense. Things should appear in an order that feels right based on everyday life

2. **What do you understand by "Single source of truth"? and how does it relate to redux? What are the advantages ?**

Answer Single Source of Truth or SSOT is a concept that means all data or information is stored in one main place. It can be seen as a model or a data schema where all important details are kept and can be linked to any part of an app.

Redux is a tool that relates to this idea. It stores all the information or data of an app in one place, making it a single source of truth. For example, all the data can be stored in a single file like store.js, which acts like a data model. Because every part of the app knows where to find this information, the app can run smoothly.

The advantages of using SSOT

- Efficiency: It takes less time to search for the correct data because all the information is in one place.

- Ease of Navigation: Information is easier to find because it is stored in a single location.

- Consistency and Reduced Errors: All data comes from one place, helping to avoid confusion or miscommunication. This reduces errors because every part of the app is referring to the same information.

- Accuracy and Up-to-Date Information: Because there is only one source of data, the information is more likely to be accurate and up-to-date."

3. What is the difference between a stateless component and a stateful component in React?

Answer In React a component is like a block of code that can be either stateful or stateless

A stateless component in React is straightforward => takes in data => processes it => shows the result. It doesn't remember anything or keep track of changes. It doesn't use a constructor, class, or hooks. Its main job is to display data or pass it down to children components, then output it in JSX form.

An example of the code

```
Function fullName(props){
Return(<h1>{props.firstName}  {props.lastName}</h1>)}
```

A stateful component in React is a piece of code that doesn't just show data, but also keeps track of it. It can collect, store, changes and display data. It utilizes lifecycle methods special functions that allow it to run code at specific points in its lifecycle. The data in a stateful component can change, be removed, or be updated, often based on user actions.

An example of the code

```
import React, { useState } from 'react';
function Subscription() {
const [email, setEmail] = useState('');
const handleSubmit = (e) => {
e.preventDefault()
}
return (
<form onSubmit={handleSubmit}>
        <input type="email" value={email} onChange={(e) => setEmail(e.target.value)}/>
         <input type="submit" value="Subscribe" />
</form>
 )}
```

4. List out the advantages and disadvantages of exploratory testing (used in Agile) and scripted testing?

Answer

Exploratory testing

Advantages

- Testers can use their skills and experience freely
- They can find bugs quickly because there's no script to follow
- It can adjust its approach based on what is learned about the system
- This testing is good for real-world conditions
- Testers can learn more about the system which leads to better testing

Disadvantages

- If a bug is found, it can be hard to find it again because there wasn't a specific path to follow
- There's usually not a lot of documentation, which can make reviewing the testing process tough
- If different testers are used, they might approach testing in different ways, and this could mean not all parts of the system get tested thoroughly

Scripted testing

Advantages

- Tests are planned and scripted, which can make it easier to find and fix bugs
- The scenarios are planned, it ensures that all parts are tested
- This is good for testers who are less experienced, as they are given specific instructions to follow

Disadvantages

- Testers must follow the scripts, which could mean they miss unexpected bugs or issues.
- It can take a lot of time and resources to create detailed test scripts.
- Sometimes these tests don't represent how the software is used in the real world. They follow a set path, rather than adapting to situations like a real user would.