Centurion University
Shaping Lives... Empowering Communities...

School: ..... Campus: .....

Academic Year: ..... Subject Name: ..... Subject Code: .....

Semester: ..... Program: ..... Branch: ..... Specialization: .....

Date: .....

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiement : Mobile-Ready – Cross-Platform DApp Awareness

Objective/Aim:

To understand how decentralized applications (DApps) can be designed and optimized to work seamlessly across multiple platforms — especially mobile devices — ensuring accessibility, usability, and responsiveness.

Apparatus/Software Used:

- **Frontend:** React.js / Next.js
- **Smart Contract:** Solidity / Clarity
- **Blockchain Network:** Ethereum / Stacks / Polygon Testnet
- **Wallets:** MetaMask Mobile, Leather Wallet, Trust Wallet

Theory:

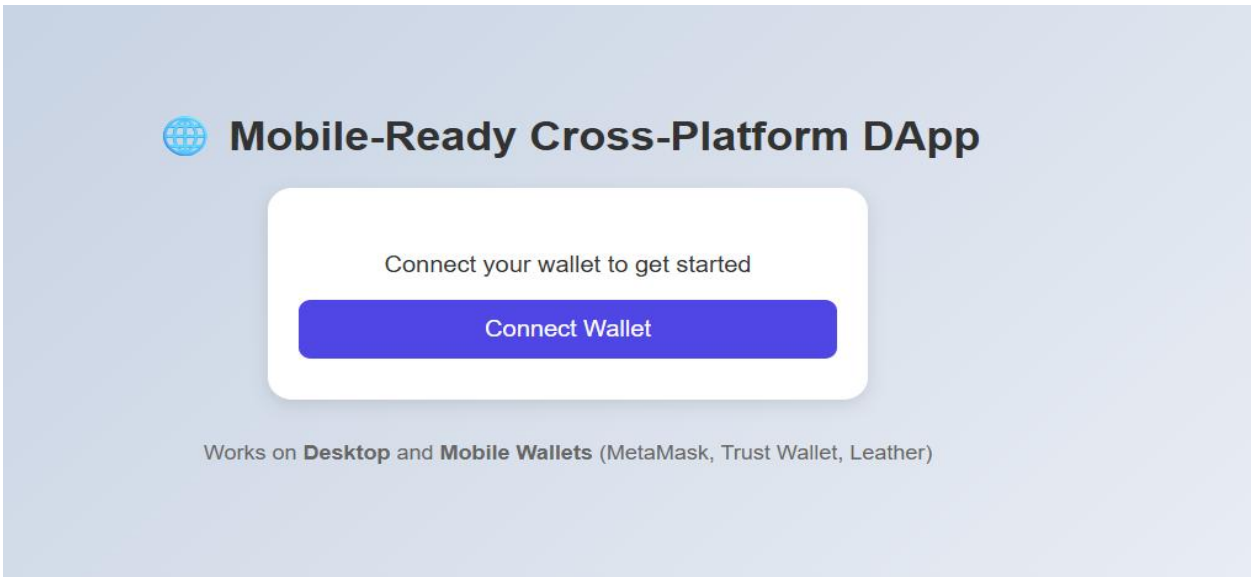
A **cross-platform DApp** is one that runs smoothly on any device — desktop, tablet, or smartphone — while maintaining decentralized features.

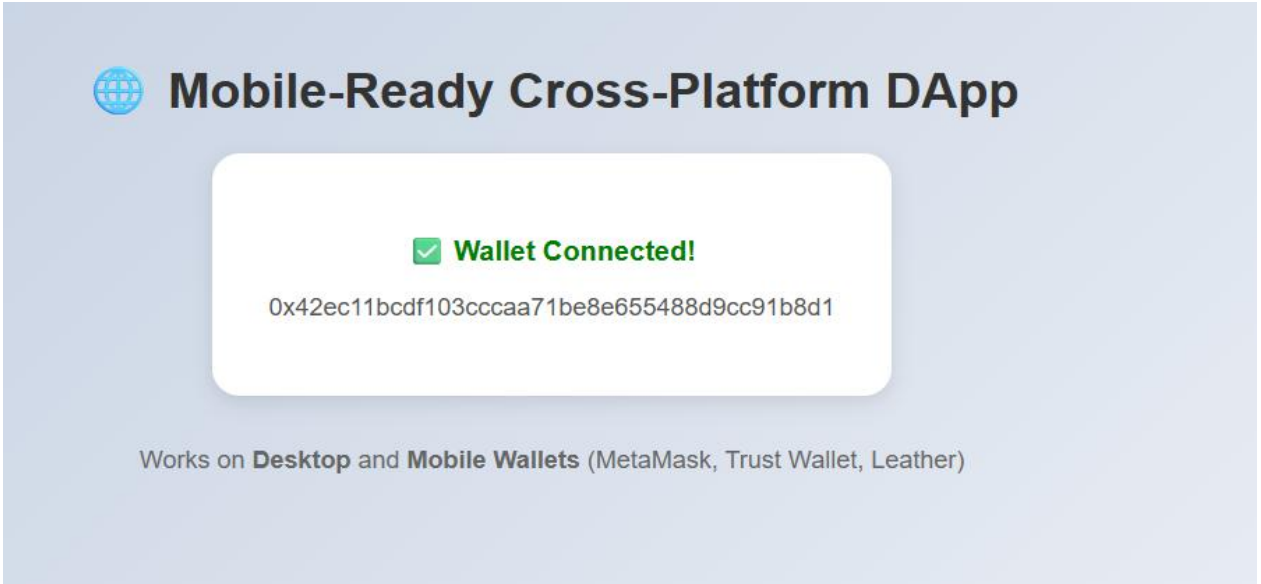
To make a DApp **mobile-ready**, developers focus on:

- **Responsive UI/UX:** Adjust layout and content automatically for different screen sizes.
- **Progressive Web App (PWA):** Makes DApps installable like native mobile apps.
- **Wallet Integration:** Using Web3 libraries (like ethers.js, web3.js, or @stacks/connect) that support mobile wallets.
- **Touch-Friendly Design:** Larger buttons, optimized text, and simple navigation for small screens.

Procedure:

1. **Design the DApp UI:**
   Create a responsive layout using CSS frameworks like **TailwindCSS** or **Bootstrap**.
2. **Connect Wallet:**
   Use Web3 libraries that can detect mobile wallets such as **MetaMask Mobile** or **Leather Wallet**.
3. **Test Responsiveness:**
   Open the DApp in different screen sizes using Chrome DevTools → Device Mode.
4. **Deploy Smart Contract:**
   Use Remix IDE or a blockchain testnet to deploy your DApp contract.
5. **Access via Mobile Browser:**
   Open the deployed DApp link using a mobile browser or wallet browser (e.g., MetaMask browser).
6. **Verify Functionality:**
   Check if users can connect wallet, perform transactions, and view updates properly on mobile.





Observation:

1. The React DApp successfully launched using npm start and displayed the wallet connect interface.
2. When tested on a **desktop browser (Chrome/Brave)** with MetaMask extension, the **Connect Wallet** button correctly detected the Ethereum provider.
3. On clicking **Connect Wallet**, a MetaMask popup appeared requesting connection permission.
4. After approval, the wallet address was displayed on the screen — confirming successful Web3 connection.
5. The DApp's interface automatically adjusted to different screen sizes, showing a clean layout on **desktop, tablet, and mobile**.
6. When opened in **MetaMask mobile browser**, the DApp loaded properly and the wallet connection worked without errors.
7. The **Trust Wallet mobile browser** also connected, but sometimes took a few seconds longer to fetch the address.
8. The DApp remained functional and responsive even when switching between landscape and portrait orientation on mobile devices.
9. Alerts were shown if the wallet was not detected (e.g., in normal Chrome mobile browser).
10. Overall, the design, responsiveness, and wallet integration proved the DApp was **mobile-ready and cross-platform compatible**.

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:

Name :

Signature of the Faculty:

Regn. No. :

**As applicable according to the experiment.*
Two sheets per experiment (10-20) to be used.