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 :

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

UI for DApps – Building a DApp Frontend :

Overall Architecture Strategy:

```
Smart Contract (Backend) ←→ Ethers.js ←→ React Frontend ←→ User
     ↓                         ↓           ↓          ↓
Blockchain State         Web3 Connection   UI Components  MetaMask
```

Technical Stack Selection :

1. Frontend Framework: React
   - Why: Most popular, extensive ecosystem, great for DApps
   - Alternative: Vue.js, Svelte, or vanilla JS
2. Web3 Library: Ethers.js
   - Why: Simpler API than web3.js, better TypeScript support
   - Key Features: Contract interaction, wallet connection, event handling
3. UI Library: Material-UI (MUI)
   - Why: Professional design, responsive, accessible components
   - Alternative: Chakra UI, Ant Design, or custom CSS
4. State Management: React Context + Hooks
   - Why: Built-in React solution, perfect for Web3 state
   - Alternative: Redux, Zustand, or Recoil

Component Architecture Design :

```
App (Root)
├── Web3Provider (Context)
├── WalletConnection (Component)
└── GreeterInterface (Component)
    └── useGreeter (Custom Hook)
```

Step 1: Project Structure Setup

```
frontend/
└── greeter-dapp/
    ├── src/
    │   ├── components/    # React components
    │   ├── context/       # Web3 state management
    │   ├── hooks/         # Custom React hooks
    │   └── config.js      # Contract addresses & ABIs
    └── public/            # Static assets
```

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

Coding Phase: Pseudo Code / Flow Chart / Algorithm

Key Design Decisions

1. Responsive Design Approach
    - Used Material-UI's grid system
    - Mobile-first responsive design
    - Accessible color contrasts and typography

2. User Experience Considerations
    - Loading states for all async operations
    - Clear feedback for transactions (pending, success, error)
    - Intuitive workflow from connection to interaction
    - Error handling with helpful messages

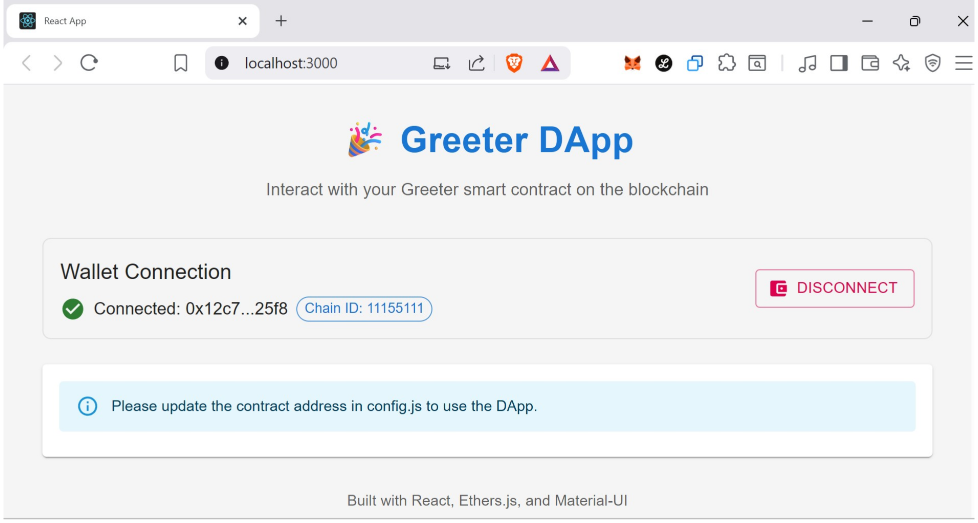
3. Developer Experience
    - Modular architecture for easy maintenance
    - Custom hooks for reusable logic
    - Configuration centralization in one file
    - Type safety with proper prop validation

* Softwares used

- **VS Code**
- **Solidity, Hardhat**
- **Javascript**
- **Ether.js**
- **Metamask**

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

* Implementation Phase: Final Output (no error)



🎉 Greeter DApp

Interact with your Greeter smart contract on the blockchain

Wallet Connection

✓ Connected: 0x12c7...25f8 Chain ID: 11155111

ⓘ Please update the contract address in config.js to use the DApp.

Built with React, Ethers.js, and Material-UI

* Observations

1. Material-UI Effectiveness:
   - Rapid development with pre-built components
   - Consistent design across application
   - Accessibility benefits out-of-the-box
   - Customization limitations for brand identity
2. Responsive Design Challenges:
   - Mobile wallet interactions different from desktop
   - Transaction details display on small screens
   - Connection status visibility across devices
   - Touch targets for critical actions

ASSESMENT

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 :

Regn. No. :

Signature of the Faculty:

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