Reasons for minimalizing Truffle usage and choosing Next.JS

This document will specify the challenges faced while MusicDapp development and how it was addressed.

1. **Deployment Challenges Using Truffle:**

   Truffle is a one stop shop for smart contract deployment, but recently the developer's community has encountered that Truffe framework is constantly at flux which gives rise to frequent code breaks on production level DApps.

   To overcome the Truffle breaking problems, I developed a node project with uses minimal truffle component for Smart Contract compilation and deployment.

   - **musicDAPP/Ethereum/compile.js:** Compiles the smart contract and returns contract bytecode and ABI (Application Binary Interface). The returned ABI and bytecodes are stored under musicDAPP/ethereum/build to avoid frequent recompilations.
   - **musicDAPP/Ethereum/deploy.js:** Reads the stored bytecode and ABI and deploys it to Rinkeby network.

2. **Developing Multi Page Dapps:**

   React JS is a popular front-end framework for developing Dapps, but it has certain limitations while developing Multipage Dapps which requires variable based routings.

   MusicDAPP requires variable based routing, once the contract is created, the show.js page will show the details of specific contract (localhost:3000/contract address/show.js).

   Since contract address is a variable which changes for every contract that's deployed, I have used next.js for support variable based routing.