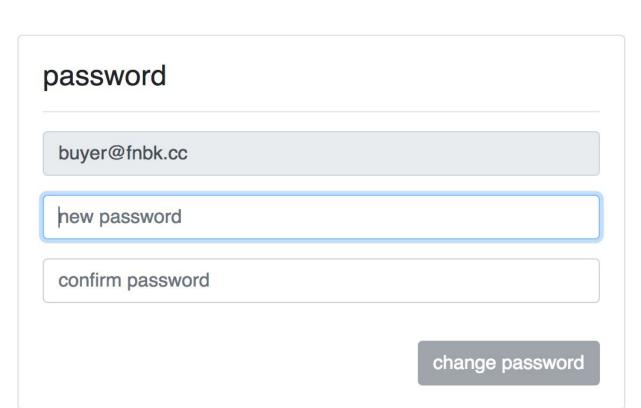
How to structure Software

ReactJS

How To Structure Software (React)

| My Account |
|----------------|
| My Watchlist |
| Buyer Requests |
| Seller Offers |
| My Contracts |
| Let's Search |



Challenge 1: Big Fat Files



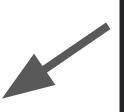
Solution 1: Old Good View & Controller

Separation of Concerns (SoC)

- Presentation
- Logic

Big Fat File

view (only props)

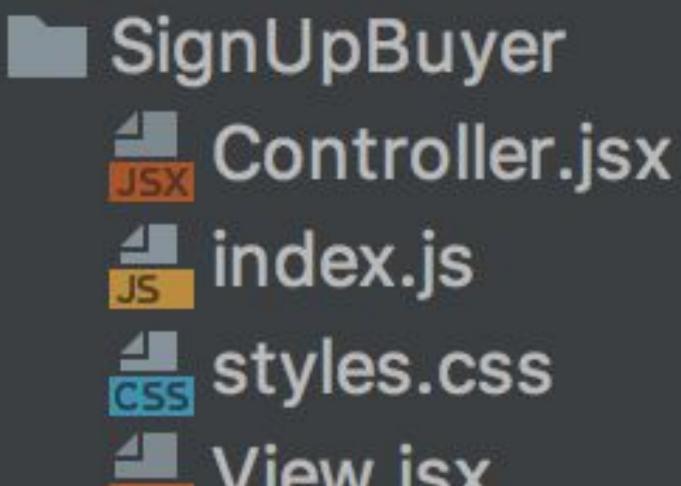


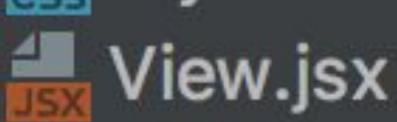


controller

```
import './styles.css';
const SignUpForm = ({
 email, password, error, onEmailChange, onPasswordChange, handleSignUp,
  <Form onSubmit={handleSignUp} className="form-centered mt-5">
    <div className="text-center mb-4">
     <h1 className="h3 mb-5 font-weight-normal">sign up buyer</h1>
    </div>
<FormGroup>
       value={email}
        type="email"
id="inputEmail"
       className="form-control"
        placeholder="Email address"
        required
    </FormGroup>
    <FormGroup>
        value={password}
       onChange={onPasswordChange}
        type="password"
       id="inputPassword"
className="form-control"
       placeholder="Password"
        required
    </FormGroup>
    <Button className="btn btn-lg btn-primary btn-block" type="submit">sign up</Button>
   { error && <Alert className="mt-2" color="secondary">{error}</Alert> }
SignUpForm.propTypes = {
  error: PropTypes.string.isRequired.
  password: PropTypes.string.isRequired,
  onPasswordChange: PropTypes.func.isRequired,
  handleSignUp: PropTypes.func.isRequired,
export default SignUpForm;
```

```
class Controller extends Component {
  constructor(props) {
     this.state = { ...INITIAL_STATE };
this.handleSignUp = this.handleSignUp.bind(this);
  async handleSignUp(e) {
  const that = this;
      const { email, password } = this.state;
      e.preventDefault();
     auth.createUserWithEmailAndPassword(email, password)
   .then(userCred => {...})
          that.setState({ error: err.message });
     const { error, email, password } = this.state;
          password={password}
         onEmailChange=(e > this.setState({ email: e.target.value }))
onPasswordChange=(e > this.setState({ password: e.target.value })}
handleSiquip=(this.handleSiquip)
Controller.propTypes = {
  history: PropTypes.object.isRequired,
```



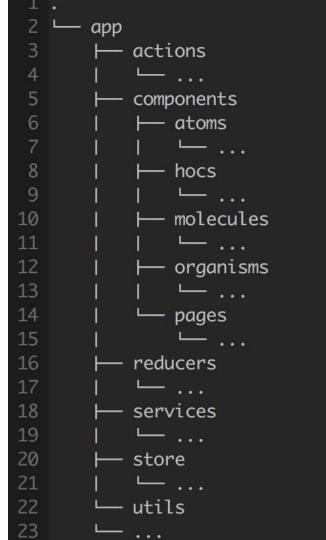


Challenge 2: File Structure

So many different names...

I don't have a clue what's going on?

Please show me the documentation!



Solution 2: Use Your Website Layout

| _ | |
|---|-------------------------|
| | 2 — Footer |
| | 3 <u> </u> |
| | 4 — Header |
| | 5 L |
| | 6 ├─ Main |
| | 7 — Account |
| | 8 Menu |
| | 9 |
| | 10 |
| | 11 |
| | 12 |
| | 13 |
| | 14 |
| | 15 ├─ Landing |
| | 16 ├── SignUpBuyer |
| | 17 — SignUpSeller |
| | 18 <u> </u> |
| | 19 |
| | |

```
const App = () => (
  <div>
    <Header />
    <Main />
    <Footer />
  </div>
```

```
const Main = () => (
  <div>
    <Switch>
      <Route exact path={SEARCH} component={Search} />
      <Route exact path={PUBLIC_PROFILE} component={PublicProfile</pre>
      <Route exact path={SIGN_UP_SELLER} component={SignUpSeller}</pre>
      <Route exact path={SIGN_UP_CUSTOMER} component={SignUpBuyer</pre>
      <Route exact path={PASSWORD_FORGET} component={PasswordForgoute</pre>
      <Route exact path={PUBLIC_PROFILE} component={PublicProfile</pre>
      <Route component={Account} />
    </Switch>
  </div>
export default Main;
```

```
const Page = () => (
          <div className="card">
            <div className="card-body">
              <Switch>
20
                <Route path={ACCOUNT} component={Profile} />
21
                <Route path={WATCHLIST} component={Watchlist} />
                <Route path={NEW REQUEST} component={NewRequest} />
                <Route path={SHOW_REQUEST} component={ShowRequest} />
                <Route path={REQUESTS} component={ListRequests} />
                <Route path={NEW OFFER} component={NewOffer} />
                <Route path={SHOW_OFFER} component={ShowOffer} />
                <Route path={OFFERS} component={ListOffers} />
31
                <Route path={NEW_CONTRACT} component={NewContract} />
32
                <Route path={SHOW_CONTRACT} component={ShowContract} />
                <Route path={CONTRACTS} component={ListContracts} />
              </Switch>
35
            </div>
          </div>
```

Question: What are your Challenges?

Challenge 3: Dependencies everywhere!

Aaaahhhh, all components are interconnected. Please help me!