

1 페이지의 이동은 리액터 라우터를 사용하여 이동한다

```
6
7  export default function RoutesSetup() {
8    return (
9      <Routes>
10        <Route path="/" element={<Login />} />
11        <Route path="/main" element={<Main />} />
12        <Route path="/login" element={<Login />} />
13        <Route path="/join" element={<Join />} />
14        <Route path="*" element={<NoMatch />} />
15      </Routes>
16    )
17  }
18
19
```

2. 첫 페이지는 로그인 페이지로 한다

3. 로그인 페이지에 조인 페이지 가기 버튼을 추가한다.



#### 4. 로그인 페이지와 조인 페이지에 유효성 검사를추가한다

```
export default function LoginContainer() {
  const [pass, setPass] = useState<string>('')
  const refEmail = useRef<HTMLInputElement>(null)
  const refPass = useRef<HTMLInputElement>(null)

  const setLabel = function (wSize: number): object {
    return {width: wSize + 'px', display: 'inline-block'}
  }

  const navigate = useNavigate()
  const goJoin = () => {
    navigate('/join')
  }

  const goLogin = () => {
    navigate('/main')
  }

  const onSubmit = (e: FormEvent<HTMLFormElement>) => {
    e.preventDefault() //웹브라우저가 onSubmit이벤트가 발생하면 form페이지를 리렌더링함, 무조건 사용.

    if (refEmail.current?.value === '') {
      alert('Please Check Email')
      if (refEmail.current !== null) refEmail.current.focus()
      return
    }
    if (refPass.current?.value === '') {
      alert('Please Check Password')
      if (refPass.current !== null) {
        refPass.current.focus()
      }
      return
    }
    goLogin()
  }

  return (
    <section
      className="set-4"
      style={{background: 'gray', width: '300px', borderRadius: '20px', color: 'white',
        title: className="mbss" style={color: "white"}}
    >

```

```
export default function JoinContainer() {
  const [wSize, setWSize] = useState<number>()
  const setWidth = (wSize: number): object => {
    return {width: wSize + 'px', display: 'inline-block'}
  }

  const onSubmit = (e: FormEvent<HTMLFormElement>) => {
    e.preventDefault() //웹브라우저가 onSubmit이벤트가 발생하면 form페이지를 리렌더링함, 무조건 사용.

    if (refId.current?.value === '') {
      alert('Please Check ID')
      if (refId.current !== null) refId.current.focus()
      return
    }
    if (refPass.current?.value === '') {
      alert('Please Check Password')
      if (refPass.current !== null) {
        refPass.current.focus()
      }
      return
    }
    if (refRPass.current?.value === '') {
      alert('Please Check RPass')
      if (refRPass.current !== null) refRPass.current.focus()
      return
    }
    if (refPass.current?.value !== refRPass.current?.value) {
      alert('password mismatch')
      if (refPass.current !== null) {
        refPass.current.focus()
      }
      return
    }
    if (refMobile.current?.value === '') {
      alert('Please Check Mobile')
      if (refMobile.current !== null) refMobile.current.focus()
      return
    }
    if (refEmail.current?.value === '') {
      alert('Please Check Email')
    }
  }

  return (
    <div
      style={{background: 'gray', width: '300px', borderRadius: '20px', color: 'white',
        title: className="mbss" style={color: "white"}}
    >

```

localhost:5173 내용:

Please Check Email

확인

localhost:5173 내용:

password mismatch

확인

### Join

ID

Pass

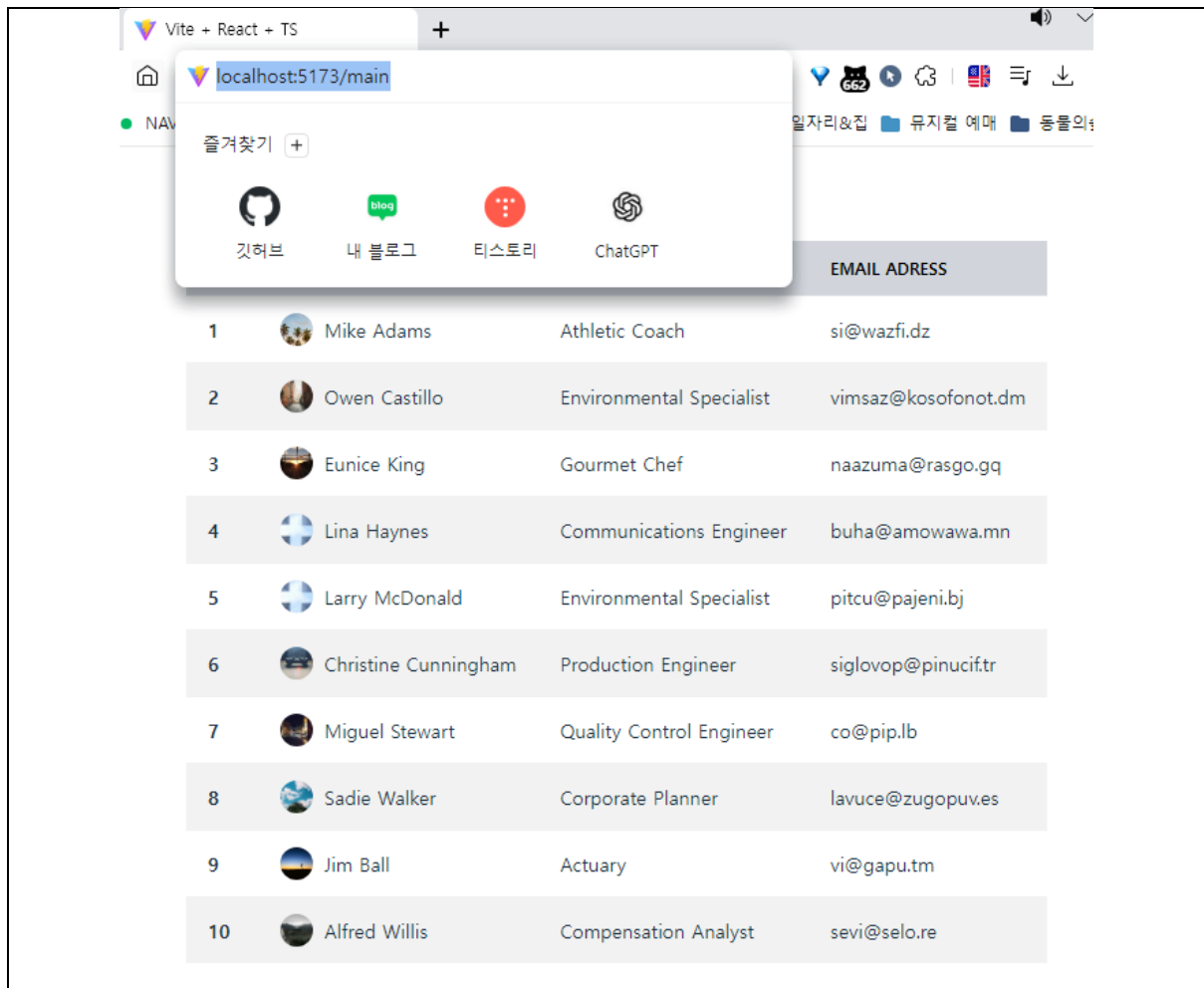
RePass

Mobile

Email

성별 ☒ 남 ☐ 여

5. 메인 페이지는 랜덤 데이터 10개를 불러와서 리스트 형식으로 보여준다



6. 프로젝트명은 ui\_이름.zip 으로 만들어서 제출하고 node\_modules는 제외하고 압축한다

7. 각 실행 장면을 캡처해서 pdf로 제출한다.