

Step 5-9: 파일 생성 및 수정

Step 5: 메인 앱 파일 설정

5-1. src/main.tsx 수정

기존 내용을 모두 삭제하고 이것으로 대체:



typescript

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.tsx'
import './index.css'

ReactDOM.createRoot(document.getElementById('root')!).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
)
```

5-2. src/App.tsx 수정

기존 내용을 모두 삭제하고 이것으로 대체:



typescript

```
import { BrowserRouter, Routes, Route } from 'react-router-dom'
import OnboardingPage from './pages/OnboardingPage'

function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/onboarding" element={<OnboardingPage />} />
        <Route path="/" element={<div className="text-center p-10">
          <h1 className="text-2xl font-bold mb-4">금융 포트폴리오 최적화 플랫폼</h1>
          <a href="/onboarding" className="text-blue-600 hover:underline">
            온보딩 시작
          </a>
        </div>} />
      </Routes>
    </BrowserRouter>
  )
}

export default App
```

5-3. src/pages/OnboardingPage.tsx 생성

새 파일 생성: src/pages/OnboardingPage.tsx



typescript

```
import { OnboardingForm } from '@components/onboarding/OnboardingForm'

export default function OnboardingPage() {
  return (
    <div className="min-h-screen bg-gradient-to-br from-blue-50 to-indigo-100 py-12">
      <div className="container mx-auto">
        <OnboardingForm />
      </div>
    </div>
  )
}
```

Step 6: .env 파일 생성

새 파일 생성: 프로젝트 루트에 .env



VITE_API_URL=http://localhost:8000

프로젝트 루트란:



portfolio-optimizer/ ← 여기
├── src/
├── package.json
├── vite.config.ts
└── .env ← 여기 생성

Step 7: API 클라이언트 수정

src/api/onboardingClient.ts 수정

기존 내용을 이것으로 대체:



typescript

```
import axios from 'axios'
import { OnboardingData } from '@types/onboarding'

const API_URL = import.meta.env.VITE_API_URL || 'http://localhost:8000'

export const onboardingAPI = {
  submit: async (data: OnboardingData) => {
    const response = await axios.post(`${API_URL}/api/onboarding/submit`, data)
    return response.data
  },

  getStatus: async (userId: string) => {
    const response = await axios.get(`${API_URL}/api/onboarding/status/${userId}`)
    return response.data
  }
}
```

Step 8: 개발 서버 시작

이미 실행 중이면 그대로 두세요.

아직 안 했으면:



bash

`npm run dev`

Step 9: TypeScript 경로 설정 (권장)

9-1. tsconfig.json 수정



json

```

{
  "compilerOptions": {
    "target": "ES2020",
    "useDefineForClassFields": true,
    "lib": ["ES2020", "DOM", "DOM.Iterable"],
    "module": "ESNext",
    "skipLibCheck": true,

    /* Bundler mode */
    "moduleResolution": "bundler",
    "allowImportingTsExtensions": true,
    "resolveJsonModule": true,
    "isolatedModules": true,
    "noEmit": true,
    "jsx": "react-jsx",

    /* Linting */
    "strict": true,
    "noUnusedLocals": true,
    "noUnusedParameters": true,
    "noFallthroughCasesInSwitch": true,

    /* Path mapping (추가) */
    "baseUrl": ".",
    "paths": {
      "@/*": ["/src/*"]
    }
  },
  "include": ["src"],
  "references": [{ "path": "./tsconfig.node.json" }]
}

```

9-2. vite.config.ts 수정



typescript

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'
import path from 'path'

export default defineConfig({
  plugins: [react()],
  resolve: {
    alias: {
      '@': path.resolve(__dirname, './src'),
    },
  },
})
```

완료 체크리스트

- ☐ Step 5-1: src/main.tsx 수정 완료
 - ☐ Step 5-2: src/App.tsx 수정 완료
 - ☐ Step 5-3: src/pages/OnboardingPage.tsx 생성 완료
 - ☐ Step 6: .env 파일 생성 완료
 - ☐ Step 7: src/api/onboardingClient.ts 수정 완료
 - ☐ Step 8: 개발 서버 실행 중
 - ☐ Step 9: TypeScript 경로 설정 완료 (선택)
-

이제 브라우저에서 확인

1. <http://localhost:5173/> 접속
 2. "온보딩 시작" 링크 클릭
 3. 첫 번째 질문 화면이 보이면 성공!
-

다음 단계

이 부분까지 완료되면:

1. **case 4-19 추가** (질문 화면)
2. **유효성 검사** (react-hook-form)
3. **페이지 레이아웃** (진행률, 버튼 등)