Next.js Fundamentals

Routing
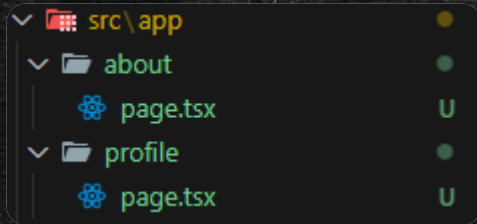
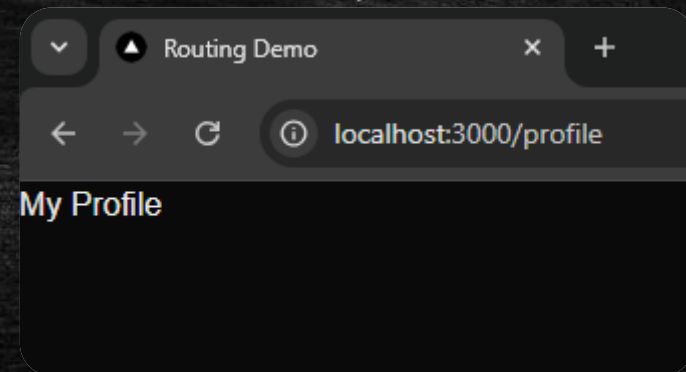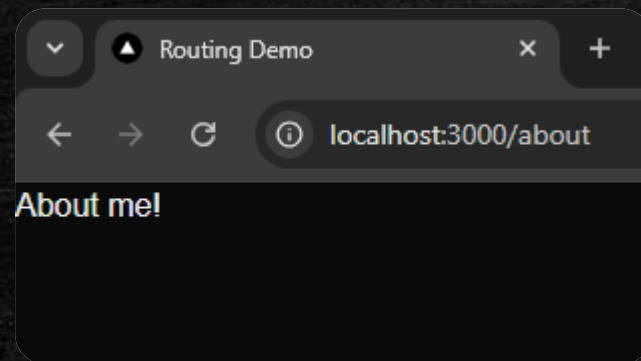npx create-next-app@latest routing-demo

Routing Conventions

- All routes must live inside the app folder

- Route files must be named either page.js or page.tsx

- Each folder represents a segment of the URL path

- When these conventions are followed, the file automatically becomes available as a route.
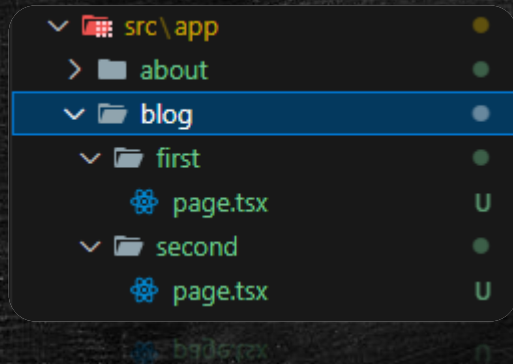
Scenario 1 (Routing)

```
src \ app
  about
    page.tsx
  profile
    page.tsx
```

```tsx
export default function About() {
  return (
    <>
      <h1>About me!</h1>
    </>
  );
}
```

```tsx
export default function Profile() {
  return (
    <>
      <h1>My Profile</h1>
    </>
  );
}
```
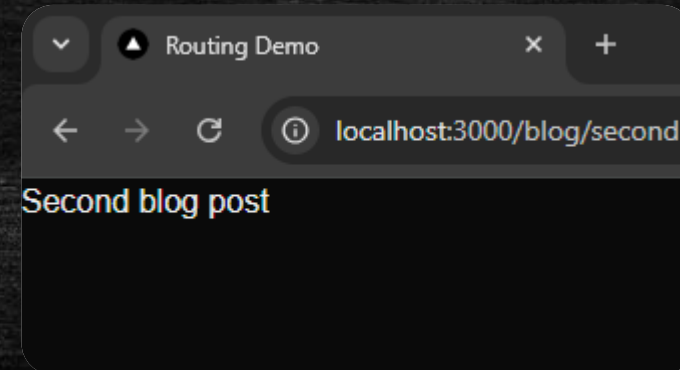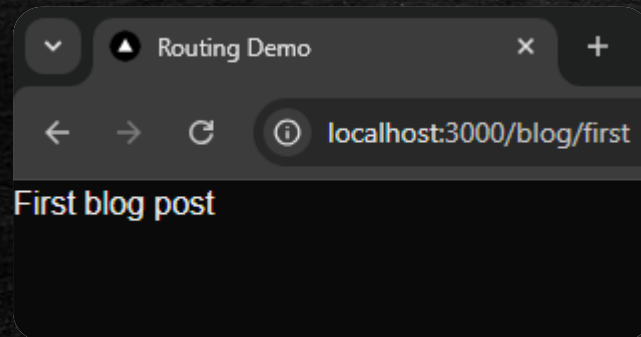
Routing Demo — localhost:3000/about

About me!

Routing Demo — localhost:3000/profile

My Profile

npm run dev

Scenario 2 (Nested Routed)

```
src\app
  about
  blog
    first
      page.tsx      U
    second
      page.tsx      U
```

```tsx
export default function FirstBlog() {
  return (
    <>
      <h1>First blog post</h1>
    </>
  );
}
```
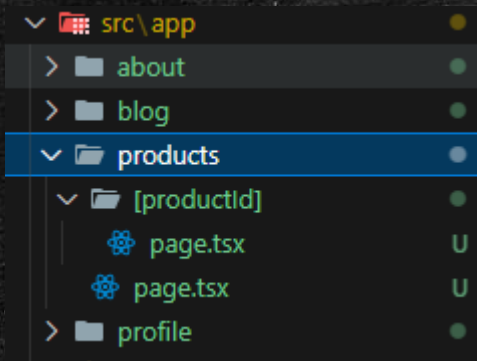
```tsx
export default function SecondPage() {
  return (
    <>
      <h1>Second blog post</h1>
    </>
  );
}
```


Routing Demo — localhost:3000/blog/first

First blog post

Routing Demo — localhost:3000/blog/second

Second blog post

Scenario 3 (Dynamic Routes)

```
src\app
  about
  blog
  products
    [productId]
      page.tsx        U
    page.tsx          U
  profile
```

```tsx
export default function Products() {
  return (
    <>
      <h1>Products List</h1>
      <ul>
        <li>Product 1</li>
        <li>Product 2</li>
        <li>Product 3</li>
      </ul>
    </>
  );
}
```
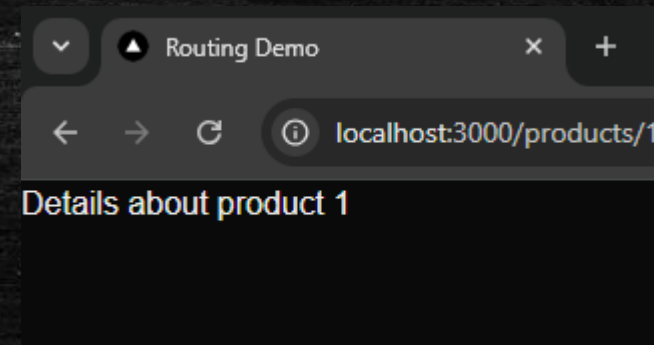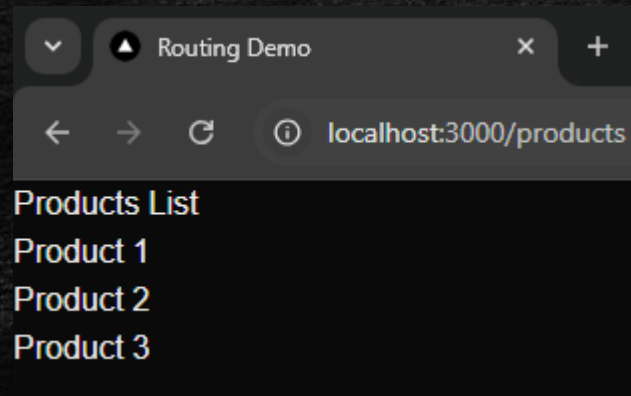
```tsx
export default async function ProductDetails({
  params,
}: {
  params: Promise<{ productId: string }>;
}) {
  const { productId } = await params;
  return (
    <>
      <h1>Details about product {productId}</h1>
    </>
  );
}
```

Routing Demo   localhost:3000/products

Products List
Product 1
Product 2
Product 3

Routing Demo   localhost:3000/products/1

Details about product 1

Scenario 4 (Nested Routes)

```
src\app
  about
  blog
  products
    [productId]
      page.tsx          U
    page.tsx            U
  profile
```

```tsx
export default function Products() {
  return (
    <>
      <h1>Products List</h1>
      <ul>
        <li>Product 1</li>
        <li>Product 2</li>
        <li>Product 3</li>
      </ul>
    </>
  );
}
```
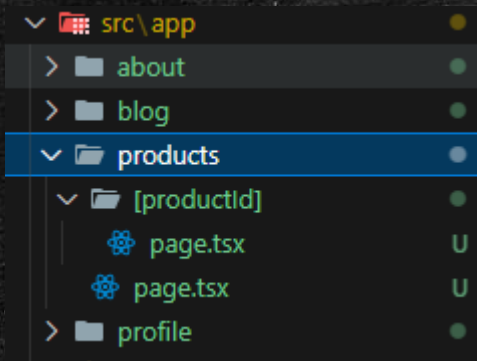
```tsx
export default async function ProductDetails({
  params,
}: {
  params: Promise<{ productId: string }>;
}) {
  const { productId } = await params;
  return (
    <>
      <h1>Details about product {productId}</h1>
    </>
  );
}
```
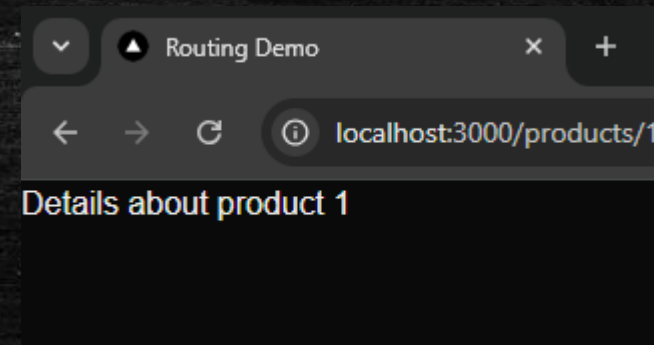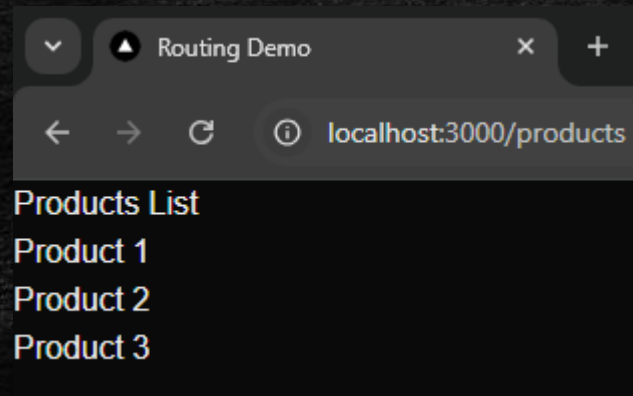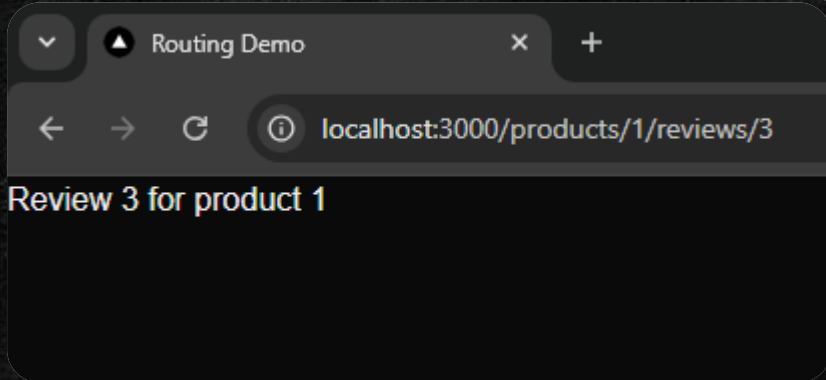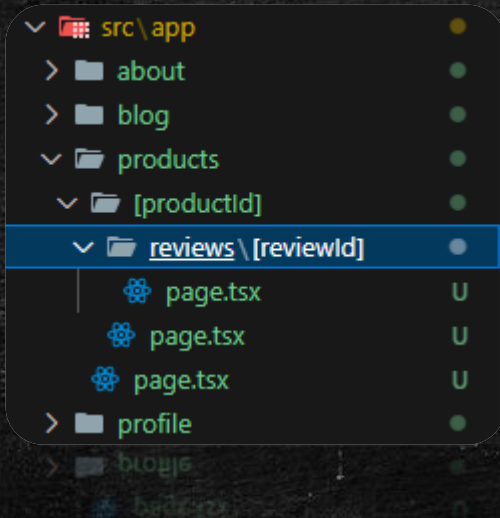
Routing Demo   localhost:3000/products

Products List

Product 1

Product 2

Product 3

Routing Demo   localhost:3000/products/1

Details about product 1

Scenario 5 (Dynamic Nested Routes)

```
src\app
  > about
  > blog
  > products
    > [productId]
      > reviews\[reviewId]
          page.tsx          U
        page.tsx            U
      page.tsx              U
  > profile
```

Routing Demo

localhost:3000/products/1/reviews/3

Review 3 for product 1

```tsx
export default async function ProductReview({
  params,
}: {
  params: Promise<{ productId: string; reviewId: string; }>;
}) {
  const { productId, reviewId } = await params;
  return (
    <>
      <h1>
        Review {reviewId} for product {productId}
      </h1>
    </>
  );
}
```

Next.js Fundamentals

Oğuzhan Dilek

https://github.com/oguzhandilek

```jsx
export default function AlbeyazimSoft() {
  return (
    <>
      <h1>The series will continue...</h1>
      <h1>Don't forget to like and share!</h1>
    </>
  );
}
```