Step 1: Create database

```
CREATE DATABASE awt_taskdb;

USE awt_taskdb;

CREATE TABLE users (

  UserID INT PRIMARY KEY AUTO_INCREMENT,

  UserName VARCHAR(50),

  Password VARCHAR(50)

);

CREATE TABLE tasks (

  TaskID INT PRIMARY KEY AUTO_INCREMENT,

  TaskTitle VARCHAR(100),

  TaskDescription TEXT,

  IsCompleted BOOLEAN,

  UserID INT,

  FOREIGN KEY (UserID) REFERENCES users(UserID)

);

INSERT INTO users (UserName, Password) VALUES

('Chirag', '1234'),

('Lallu', '5678');


INSERT INTO tasks (TaskTitle, TaskDescription, IsCompleted, UserID) VALUES

('DB Assignment', 'Complete MySQL lab', false, 1),

('AWT Project', 'Build Prisma app', true, 1),

('Testing', 'Unit testing', false, 2);
```

Step 2: Create Project

```
npx create-next-app awt-lab12

cd awt-lab12
```

Step 3: Install prisma using following command

```
npm i prisma --save-dev
```

npm i @prisma/client

npm i @prisma/adapter-mariadb

create .env file: DATABASE_URL="mysql://root@localhost:3306/awt_taskdb"

Step 4: npx prisma init

npx prisma db pull (compare number of modals pulled with your db tabels)

npx prisma generate

Step 5: create prisma file (lib/prisma.ts)

```
import { PrismaMariaDb } from "@prisma/adapter-mariadb";
import { PrismaClient } from "./generated/prisma/client";

const adapter = new PrismaMariaDb({
    host:"localhost",
    port:3306,
    user:"root",
    password:"",
    database:"awt_taskdb",
    connectionLimit:5
})

export const prisma = new PrismaClient({adapter});
```

Step 6: create a new folder named users (app/users) and then create page.tsx file.

```
import { users } from '@/lib/generated/prisma/browser';
import { prisma } from '@/lib/prisma'
import React from 'react'

async function UserList() {
    const data = await prisma.users.findMany();

  return (
    <div>
        <ul>
      {
        data.map((u:users)=>(
            <li key={u.UserID}>{u.UserName}</li>
        ))
      }
      </ul>
    </div>
  )
}

export default UserList
```

//check the output

Step 7: create a new folder named [id] inside users (app/users/[id]) and then create page.tsx file.

```
import { prisma } from '@/lib/prisma'
import Link from 'next/link';
import React from 'react'

async function DetailUser({params}:{params:Promise<{id:number}>}) {
    const {id} = await params;
    const data = await prisma.users.findFirst({where:{UserID:Number(id)}})
  return (
    <div>
      <h1>{data?.UserName}</h1>
      <Link href={"/users"}>Back</Link>
    </div>
  )
}

export default DetailUser
```

//check the output

Step 8: Create a server action (/app/actions/DeleteUserAction.tsx)

```
"use server"

import { prisma } from "@/lib/prisma";
import { revalidatePath } from "next/cache";
import { redirect } from "next/navigation";

async function DeleteUserAction(id:number){
    await prisma.users.delete({where:{UserID:id}});
    revalidatePath("/users");
    redirect("/users");
}

export {DeleteUserAction};
```

Step 9: Create a deleteui (/app/ui/DeleteButton.tsx)

```
"use client"
import React from 'react'
import { DeleteUserAction } from '../actions/DeleteUserAction'
```

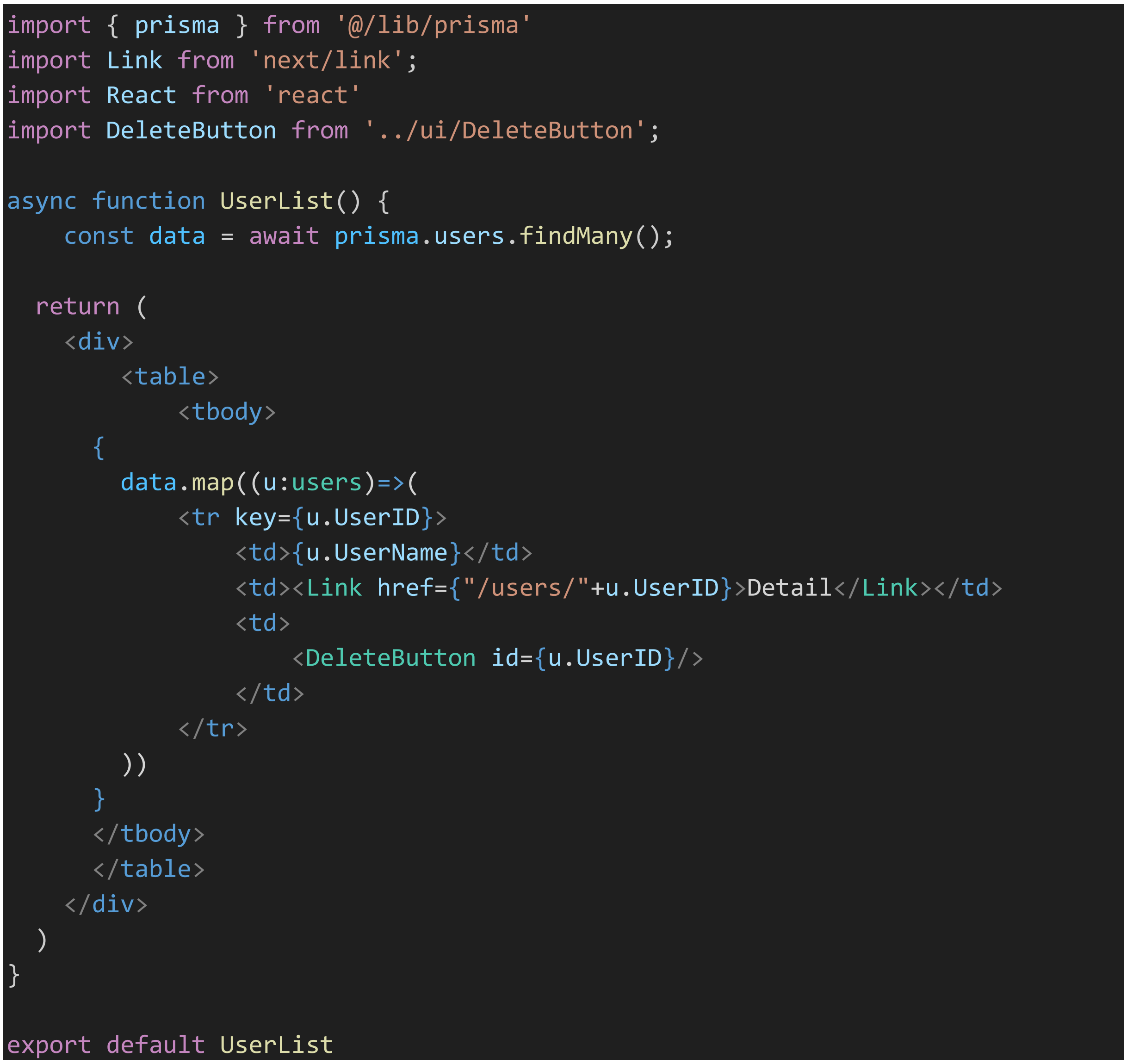
```
function DeleteButton(paramas:any) {
    const {id} = paramas;
  return (
    <button onClick={()=>{
        DeleteUserAction(id)
    }}>Delete</button>
  )
}

export default DeleteButton
```

Step 10: Add Delete UI in list page (/api/users/page.tsx)

```
import { users } from '@/lib/generated/prisma/browser';
import { prisma } from '@/lib/prisma'
import Link from 'next/link';
import React from 'react'
import DeleteButton from '../ui/DeleteButton';

async function UserList() {
    const data = await prisma.users.findMany();

  return (
    <div>
        <table>
            <tbody>
      {
        data.map((u:users)=>(
            <tr key={u.UserID}>
                <td>{u.UserName}</td>
                <td><Link href={"/users/"+u.UserID}>Detail</Link></td>
                <td>
                    <DeleteButton id={u.UserID}/>
                </td>
            </tr>
        ))
      }
      </tbody>
      </table>
    </div>
  )
}

export default UserList
```

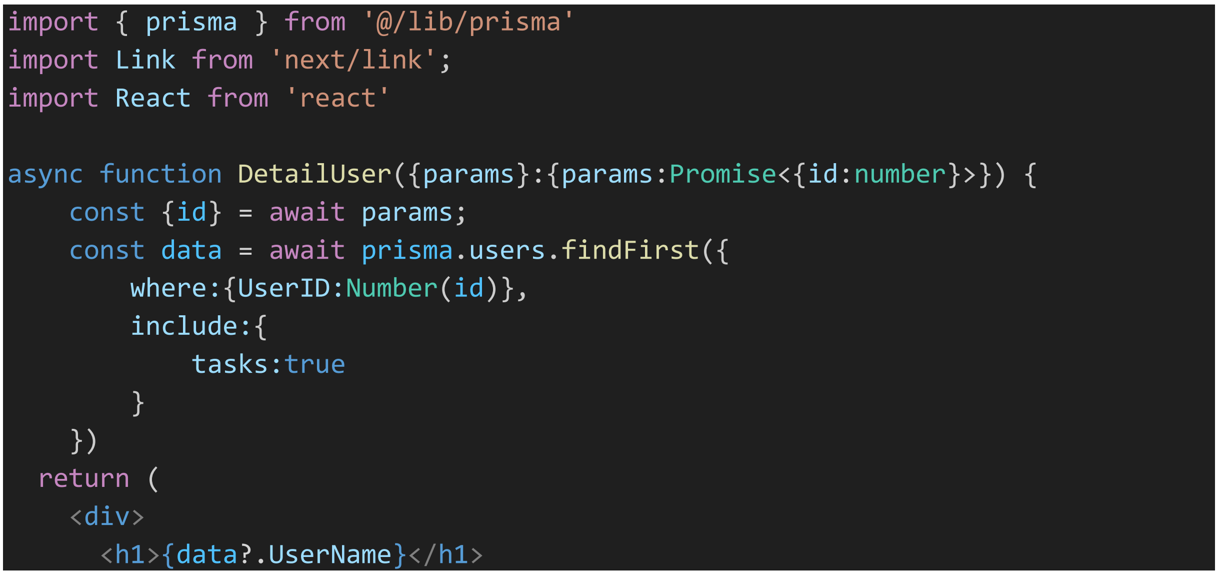
Step 11: Modify the code of (/app/users/page.tsx)

```
import { users } from '@/lib/generated/prisma/browser';
```



```
import { prisma } from '@/lib/prisma'
import Link from 'next/link';
import React from 'react'
import DeleteButton from '../ui/DeleteButton';

async function UserList() {
    const data = await prisma.users.findMany();

  return (
    <div>
        <table>
            <tbody>
      {
        data.map((u:users)=>(
            <tr key={u.UserID}>
                <td>{u.UserName}</td>
                <td><Link href={"/users/"+u.UserID}>Detail</Link></td>
                <td>
                    <DeleteButton id={u.UserID}/>
                </td>
            </tr>
        ))
      }
      </tbody>
      </table>
    </div>
  )
}

export default UserList
```

Step 12: Modify the code of (/app/users/[id]/page.tsx)



```
import { prisma } from '@/lib/prisma'
import Link from 'next/link';
import React from 'react'

async function DetailUser({params}:{params:Promise<{id:number}>}) {
    const {id} = await params;
    const data = await prisma.users.findFirst({
        where:{UserID:Number(id)},
        include:{
            tasks:true
        }
    })
  return (
    <div>
      <h1>{data?.UserName}</h1>
      <ul>
        {
            data?.tasks.map((t:any)=>(
                <li key={t.TaskID}>{t.TaskTitle}</li>
            ))
        }
      </ul>
      <Link href={"/users"}>Back</Link>
    </div>
  )
}

export default DetailUser
```

Step 13: Create a server action to insert data in the table as

(/app/actions/AddUserAction.tsx)

```
"use server"

import { prisma } from "@/lib/prisma";
import { revalidatePath } from "next/cache";
import { redirect } from "next/navigation";

async function AddUserAction(formData: FormData){
    // console.log(formData)

    const UserName = formData.get("UserName") as string;
    const Password = formData.get("Password") as string;

    const data = {UserName, Password};

    await prisma.users.create({data});

    revalidatePath("/users");
    redirect("/users")
}

export {AddUserAction}
```

Step 14: Create a form (/app/users/add/page.tsx)

```
import { AddUserAction } from '@/app/actions/AddUserAction'
import React from 'react'

function AddUser() {
  return (
    <div>
      <form action={AddUserAction}>
```

```
        <input type='text' name="UserName"/>
        <input type='text' name="Password"/>
        <input type='submit'/>
      </form>
    </div>
  )
}

export default AddUser
```