ДОДАТОК А

Код для завантаження файлів

Лістинг А.1 – Програмний код файлу UploadFile.tsx

```
"use client";
import React, { useState } from "react";
import {
 Dialog,
 DialogContent,
 DialogDescription,
 DialogHeader,
 DialogTitle,
 DialogTrigger,
} from "@/components/ui/dialog";
import {
 Form,
 FormControl,
 FormField,
 FormItem,
 FormLabel,
 FormMessage,
} from "@/components/ui/form";
import { Button } from "@/components/ui/button";
import { Upload } from "lucide-react";
import { Input } from "@/components/ui/input";
import { zodResolver } from "@hookform/resolvers/zod";
import { useForm } from "react-hook-form";
import { z } from "zod";
import { axiosInstance } from "@/services/axios";
import { useToast } from "@/components/ui/use-toast";
import { FilesFavorite } from "@/types/types";

const formSchema = z.object({
 file: z
   .custom<FileList>((val) => val instanceof FileList,
"Required")
   .refine((files) => files.length > 0, `Required`),
});

const UploadFile = ({
 setFiles,
}: {
 setFiles:
React.Dispatch<React.SetStateAction<FilesFavorite[] | []>>;
}) => {
 const [open, setOpen] = useState<boolean>(false);

 const form = useForm<z.infer<typeof formSchema>>({
   resolver: zodResolver(formSchema),
   defaultValues: {
```

Продовження лістингу А.1

```
        file: undefined,
      },
    });

    const { toast } = useToast();
    const fileRef = form.register("file");

    const onSubmit = async (values: z.infer<typeof
formSchema>) => {
      const formData = new FormData();
      formData.append("file", values.file[0]);
      const config = {
        headers: {
          "Content-Type": "multipart/form-data",
        },
      };

      try {
        const response = await axiosInstance.post(
          "/api/v1/file/upload",
          formData,
          config,
        );
        if (response.status === 200) {
          setFiles((prevState) => [
            ...prevState,
            { data: response.data, fav: null },
          ]);
          toast({
            title: "File Uploaded",
          });
          setOpen(false);
        }
      } catch (e) {
        toast({
          title: "Something went wrong",
          description: "Your file could not be uploaded, try
again later",
        });
      }
    };

    return (
      <Dialog open={open} onOpenChange={setOpen}>
        <DialogTrigger asChild>
          <Button variant="outline" className="text-[15px]">
            <Upload width={18} className="mr-2" />
            Upload file
          </Button>
        </DialogTrigger>
        <DialogContent className="sm:max-w-[425px]">
```

Продовження лістингу А.1

```
            <DialogHeader>
              <DialogTitle>Upload a file</DialogTitle>
              <DialogDescription>
                Make sure that your file is less than 21MB.
              </DialogDescription>
            </DialogHeader>
            <Form {...form}>
              <form onSubmit={form.handleSubmit(onSubmit)}
className="w-full">
                <FormField
                  control={form.control}
                  name="file"
                  render={({ field }) => (
                    <FormItem>
                      <FormLabel>File</FormLabel>
                      <FormControl>
                        <Input type="file" {...fileRef} />
                      </FormControl>
                      <FormMessage />
                    </FormItem>
                  )}
                />
                <Button type="submit" className="mt-4 w-full">
                  Upload
                </Button>
              </form>
            </Form>
          </DialogContent>
        </Dialog>
      );
    };

    export default UploadFile;
```

ДОДАТОК Б

Код компоненти Dashboard

Лістинг Б.1 – Програмний код файлу SearchPage.tsx

```
"use client";
import React, { useState, useEffect } from "react";
import { axiosInstance } from "@/services/axios";
import { useAuth } from "@/services/auth/AuthProvider";
import { FilesFavorite } from "@/types/types";
import FileCards from "@/components/FileCards";
import Header from "@/components/Header";

const Dashboard = () => {
 const { user } = useAuth();

 const [files, setFiles] = useState<FilesFavorite[] |
[]>([]);
 const [filesLoading, setFilesLoading] =
useState<boolean>(true);

 useEffect(() => {
   try {
     const fetchFiles = async () => {
       const response =
         await
axiosInstance.get<FilesFavorite[]>("/api/v1/files");
       if (response.status === 200) {
         setFiles(response.data);
       }
     };
     fetchFiles().then();
   } catch (e) {
     console.log("Error with file request.");
   } finally {
     setFilesLoading(false);
   }
 }, []);

 return (
   <>
     <div className="my-8">
       <Header
         title="Your Files"
         setFiles={setFiles}
         setLoading={setFilesLoading}
       />
     </div>
     <div className="grid grid-cols-5 gap-5">
       {filesLoading ? (
```

Продовження лістингу Б.1

```
              <div className="col-span-4 h-[calc(100vh-300px)]
flex justify-center items-center">
                <div className="loader"></div>
              </div>
            ) : files.length ? (
              files.map((file, i) => (
                <React.Fragment key={i}>
                  <FileCards
                    user={user}
                    file={file.data}
                    fav={!!file.fav}
                    files={files}
                    setFiles={setFiles}
                  />
                </React.Fragment>
              ))
            ) : (
              <div className="col-span-4 h-[calc(100vh-300px)]
flex justify-center items-center">
                <img
                  className="w-[450px] h-auto rounded-full"
                  src="/no-data.png"
                  alt=" "
                />
              </div>
            )}
          </div>
        </>
      );
    };
    export default Dashboard;
```

ДОДАТОК В

Код провайдера для автентифікації

Лістинг В.1 – Програмний код файлу AuthProvider.tsx

```
import React, { createContext, useContext, useState,
useEffect } from "react";
import { axiosAuthInstance } from "@/services/axios";
import { User } from "@/types/types";

interface AuthContextType {
 user: User | null;
 loading: boolean;
 login: (userData: User) => void;
 logout: () => void;
}

const AuthContext = createContext<AuthContextType |
undefined>(undefined);

export const AuthProvider = ({ children }: { children:
React.ReactNode }) => {
 const [user, setUser] = useState<User | null>(null);
 const [loading, setLoading] = useState<boolean>(true);

 useEffect(() => {
   const token = localStorage.getItem("access_token");
   try {
     if (token) {
       const fetchUser = async () => {
         const user = await
axiosAuthInstance.get<User>("/api/v1/user");
         if (user.status === 200) {
           setUser(user.data);
         }
       };
       fetchUser().then();
     }
   } catch (e) {
     console.log("Error with user request.");
   } finally {
     setLoading(false);
   }
 }, []);

 const login = (userData: User) => {
   setUser(userData);
 };

 const logout = async () => {
   setUser(null);
```

Продовження лістингу В.1

```
        const token = localStorage.getItem("access_token");
        await axiosAuthInstance.post("/api/v1/logout", {
access_token: token });
        localStorage.removeItem("access_token");
      };

      const authContextValue: AuthContextType = {
        user,
        loading,
        login,
        logout,
      };

      return (
        <AuthContext.Provider value={authContextValue}>
          {children}
        </AuthContext.Provider>
      );
     };

     export const useAuth = (): AuthContextType => {
      const context = useContext(AuthContext);
      if (!context) {
        throw new Error("useAuth must be used within an
AuthProvider");
      }
      return context;
     };
```

ДОДАТОК Г

Відомість кваліфікаційної роботи

Позначення	Найменування	Дод. відомості
	Текстові документи	
1.	Пояснювальна записка	67 с.
	Інші документи	
2.	Презентаційні матеріали	12 плакатів

.	.	Прізвище та ініціали.	Підп.	Дата	
Розробив		Коваль Г. К.			Оптимізація пошуку в файлових сховищах засобами NLP моделей
Перевірив		Гриньова О.Є			
Н.контр.		Малєєва І.А.			
Затв.		Золотухін О.В.			

Шифр групи	Код напр./спец.
ІТШІ-21-2	122
ХНУРЕ кафедра ШІ	