

Додаток А
(Рекомендований)

ПРОГРАМНІ КОДИ

Код верхньої панелі (Розмітка та підключення елементів для верхньої панелі)

```

"use client";
import Link from "next/link";
import { useSession } from "next-auth/react";
import { usePathname } from "next/navigation";
import { btnColorSchemas }
from "@/components/buttons/ButtonClassNamerSchemas";
import Image from "next/image";
import { BsFillPersonFill, BsGear } from "react-icons/bs";
import { SearchBar } from "./inputs/SearchBar";

type NavLink = { id: string; label: string; href: string };
type Props = {
  navLinks: NavLink[];
  isPlayerLayout?: boolean;
  className?: string;
};

const Navigation = ({ navLinks, isPlayerLayout, className }: Props) => {
  const pathname = usePathname();
  const { data: session, status, update } = useSession();

  return (
    <nav className={` ${className} w-full m-0`}>
      <div
        className={` flex items-center justify-between ${isPlayerLayout ? "md:justify-end" : ""}
          } py-[18px] px-[30px] h-24`
        >
        <Link href="/" className={` ${isPlayerLayout ? "md:hidden" : ""}`}>
          <Image src="/Mmusic.png" alt="Music Logo" width="250" height="46" />
        </Link>
        <div className="w-full flex justify-end gap-x-12">
          {isPlayerLayout && <div className="w-[240px] pointer-events-none"></div>}
          {!isPlayerLayout && (
            <div className="flex items-center gap-x-8">
              {navLinks.map((link, idx) => (
                <Link
                  key={`hd.${idx}.${link.id}`}
                  href={link.href}
                  className={` ${pathname === link.href ? "text-green" : ""}
                    } hover:text-green`}
                >
                  {link.label}
                </Link>
              ))}
            </div>
          )}
        </div>
      </div>
      <label className="text-[22px] font-bold w-max flex items-center justify-end gap-x-5"
        for="language">Оберіть мову:</label>

```

```

<select id="language" name="language">
  <option className="text-[22px] font-bold" value="ukrainian">Українська</option>
  <option className="text-[22px] font-bold" value="english">English</option>
</select>
{isPlayerLayout && <SearchBar />}
<div className="w-max flex items-center justify-end gap-x-5">
  {session ? (
    <>

      <Link
        href="/profile"
        className={`flex justify-center items-center rounded-full w-[35px] h-[35px]
overflow-hidden ${btnColorSchemas.white}`}
      >
        {session?.user?.image ? (
          <img
            src={session?.user?.image}
            alt={`Avatar ${session?.user?.name}`}
          />
        ) : (
          <BsFillPersonFill size={24} />
        )}
      </Link>
    </>
  ) : (
    <Link
      href="/signin"
      className={`font-bold border rounded-full py-2 px-4 ${btnColorSchemas.violet}`}
    >
      <span className="text-[22px] font-bold w-max block">Реєстрація</span>
    </Link>
  )}
  <Link
    href="/signin"
    className={`text-[20px] font-bold border rounded-full py-2 px-4
${btnColorSchemas.violet}`}
  >
    <span className="text-[22px] font-bold w-max block">Увійти</span>
  </Link>
  <Link
    href="/profile/plans"
    className={`border rounded-full py-3 px-2 ${btnColorSchemas.black}`}
  >
    <span className="text-[20px] font-bold w-max block">Допомога</span>
  </Link>
  <Link
    href="/profile/plans"
    className={`border rounded-full py-3 px-2 ${btnColorSchemas.black}`}
  >
    <span className="text-[20px] font-bold w-max block">Приватність</span>
  </Link>
  <Link

```

```
      href="/profile/plans"
      className={`border rounded-full py-3 px-2 ${btnColorSchemas.black}`}
    >
      <span className="text-[20px] font-bold w-max block">Умови</span>
    </Link>
  </div>
</div>
</div>
</nav>
);
};

export { Navigation };
```

Код бокової панелі**(Розмітка та підключення елементів для бокової панелі основної сторінки)**

```

"use client";
import { useBearStore } from "@store";
import { clsxMerge } from "@utils/clsxmerge";
import Image from "next/image";
import Link from "next/link";
import { usePathname } from "next/navigation";
import { useEffect, useState } from "react";
import { BsFillPersonFill, BsGear } from "react-icons/bs";
import {
  BsHeart,
  BsHouseDoor,
  BsPlusCircle,
  BsSearch,
  BsStack,
} from "react-icons/bs";

const playlists = [
  { label: "Energetic", href: "/player/playlist/16JGZG9gKRs6jtbz2kb80b" },
  { label: "Calm", href: "/player/playlist/71P9dVzRbxGVWw21G0UrV8" },
  { label: "Motivated", href: "/player/playlist/5bevjRB0KTt4KIXdFhB141" },
  { label: "Happy", href: "/player/playlist/37i9dQZF1DXdPec7aLTmlC" },
  { label: "Reflective", href: "/player/playlist/37i9dQZF1EIdUxgjQEPCoS" },
  { label: "Inspirational", href: "/player/playlist/5bUrUEjOpSgpTUZ4zcAPqn" },
  { label: "Romantic", href: "/player/playlist/4RaGFH3E9PZhe8YhhJ3ao2" },
  { label: "Adventurous", href: "/player/playlist/0suj0O532SJW9D3bp6Xd0u" },
  { label: "Empowered", href: "/player/playlist/5BxvtX6Vv3thSeaiMIlc77" },
  { label: "Nostalgic", href: "/player/playlist/2wr1SGXUphkqYPKbyYiwbY" },
  { label: "Serene", href: "/player/playlist/3KzfDaSGyQJBZITNxcI89r" },
  { label: "Mysteriuos", href: "/player/playlist/37i9dQZF1DZ06evO0ZJfqh" },
  { label: "Upbeat", href: "/player/playlist/56iNGOSGtoHthCJqltVHdq" },
  { label: "Melancholic", href: "/player/playlist/0aj82CtYGkX76DzBYFJC1V" },
  { label: "Excited", href: "/player/playlist/0BL7LYpkYb7mIWizLc58pC" },
  { label: "Relieved", href: "/player/playlist/4hYvLg7dMuQLTLcu8IHJxs" },
  { label: "Playful", href: "/player/playlist/0wAVH11eJWOprYBmsMeLMP" },
  { label: "Confident", href: "/player/playlist/0eJ60UuuU19trAaCI51rkH" },
  { label: "Sentimental", href: "/player/playlist/37i9dQZF1EIgT2G8KHqjrj" },
  { label: "Mystical", href: "/player/playlist/1krltTJ8IO8WRQZhrv3ZZ0" },
];

const TheSidebar = () => {
  const pathname = usePathname();

  return (
    <aside className="hidden phone:block fixed top-0 left-0 w-[200px] h-screen">
      <div
        className={`flex flex-col space-y-[46px] h-full pl-5 pr-1 pt-6 pb-36 overflow-y-auto text-grey bg-backgroundGrey`}
      >

```

```

<Link href="/" className="px-1">
  <Image src="/Mmusic.png" alt="Mmusic Logo" width="75" height="35" />
</Link>
<div className="flex flex-col space-y-4 font-bold text-base">
  <Link
    href="/player"
    className={`flex items-center hover:text-white${
      pathname === "/player" ? " text-white" : ""
    }}`
  >
    <BsHouseDoor />
    <span className="ml-3">Home</span>
  </Link>

  <Link href="#" className="flex items-center hover:text-white">
    <BsSearch />
    <span className="flex-1 ml-3 whitespace-nowrap">Search</span>
  </Link>

  <Link href="/player/me" className="flex items-center hover:text-white">
    <BsStack />
    <span className="flex-1 ml-3 whitespace-nowrap">Your Library</span>
  </Link>
</div>
<div className="flex flex-col space-y-4 font-bold text-base">
  <Link href="#" className="flex items-center hover:text-white">
    <BsPlusCircle />
    <span className="flex-1 ml-3 whitespace-nowrap">
      Create Playlist
    </span>
  </Link>

  <Link href="/player/me" className="flex items-center hover:text-white">
    <BsHeart />
    <span className="flex-1 ml-3 whitespace-nowrap">Liked Songs</span>
  </Link>
</div>
<Link className="w-max" href="/profile/settings">
  <BsGear
    size={24}
    className={` ${pathname === "/profile/settings" ? "text-green" : ""
    } hover:text-green` }
  />
</Link>{" "}
<div className="flex flex-col font-medium text-base">
  {playlists.map((playlist) => (
    <Link
      key={playlist.href}
      href={playlist.href}
      className={clsxMerge(
        `transition-all leading-5 py-[15px] hover:px-[15px] hover:font-semibold hover:bg-mint
        hover:text-black`,

```

```
    {
      "p-[15px] font-semibold bg-mint text-black":
        pathname === playlist.href,
    }
  ))
>
  {playlist.label}
</Link>
))
</div>
</div>
</aside>
);
};

export { TheSidebar };
```

Код пошуку**(Пошук по компонентам назви треків, альбомів та ніків артистів по всій системі)**

```
'use client'
```

```
import { CardRow } from "@components/CardRow";
import { useSearch } from "@components/hooks/useSearch";
import { CardRowPlaceholder } from "@components/palceholder/CardRowPlaceholder";
import { useRouter, useSearchParams } from "next/navigation"
import { useEffect, useState } from "react";
```

```
let quitTimeout: NodeJS.Timeout | null = null;
```

```
export default function Home () {
  const [result, setResult] = useState<any>();
  const router = useRouter();
  const { query } = useSearch();

  const fetchData = async () => {
    const res = await fetch(`/api/search?q=${query}&limit=5`)
    const userRes = await fetch(`/api/search/users?q=${query}`)
    const data = await res.json();
    const userData = await userRes.json();
```

```
    const fetched = {
      ...data,
      users: [...userData]
    }
    console.log(fetched)
    setResult(fetched);
  }
}
```

```
useEffect(() => {
  console.log(query)
  if(query == undefined || query == "") {
    quitTimeout = setTimeout(() => {
      router.replace('/player/');
    }, 5000);
    return;
  }
}
```

```
  setResult(null);
  fetchData();
}, [query]);
```

```
const playlistsFound = result?.playlists?.items?.length > 0;
const albumsFound = result?.albums?.items?.length > 0;
const tracksFound = result?.tracks?.items?.length > 0;
const artistsFound = result?.artists?.items?.length > 0;
const usersFound = result?.users?.length > 0;
```

```

const sthFound = playlistsFound || albumsFound || tracksFound || artistsFound || usersFound;

return (
  <>
    {result !== null && <>
      {sthFound ? <div className="flex flex-col gap-[30px]">
        {usersFound && <CardRow
          header="Users"
          rawData={result?.users}
        />}
        {playlistsFound && <CardRow
          header="Playlists"
          rawData={result?.playlists?.items}
        />}
        {albumsFound && <CardRow
          header="Albums"
          rawData={result?.albums?.items}
        />}
        {tracksFound && <CardRow
          header="Tracks"
          path="single"
          imgPath="album"
          rawData={result?.tracks?.items}
        />}
        {artistsFound && <CardRow
          header="Artists"
          rawData={result?.artists?.items}
        />}
      </div> :
      <div className="w-full flex justify-center pb-[65vh]"><span className="text-8xl
font-bold py-8">NOTHING FOUND</span></div>}
    </>
    {result === null && <>
      <CardRowPlaceholder
        header="Playlists"
      />
      <CardRowPlaceholder
        header="Albums"
      />
    </>
  </>
)
}

```

Основний код роботи чату із сумісним прослуховуванням (Обробка запитів на змінення станів сумісної пісні)

```

import { NextRequest, NextResponse } from "next/server";
import { connectDB } from "@/libs/mongodb";
import roomModel from "@/app/models/room";
import { redirect } from "next/navigation";
import { getSpotify } from "@/libs/spotify";

export async function PUT(
  req: NextRequest,
) {
  const { _id, userId, queue } = await req.json();

  await connectDB();

  try {
    const data = await roomModel.findById(_id);

    if(data == null) return NextResponse.json({
      msg: "NOT FOUND"
    });

    if(data?.members?.find((user: any) => user?.userId == userId) == null) {
      return NextResponse.json({
        error: "No Access",
        path: "/player/room/invite"
      });
    }

    if(data.adminId != userId && !data.isFreeForAll) {
      return NextResponse.json({
        msg: "Not Admin",
      });
    }

    console.log(`${queue.isPlaying ? "" : "NOT"} PLAYING T:${queue.timeCode}`);
    data.queue = queue;
    await data.save();

    return NextResponse.json('ok');
  } catch (error) {
    console.log(error)
    return NextResponse.json(error);
  }
}

export async function GET(
  req: NextRequest,
) {
  const _id = req.nextUrl.searchParams.get("_id")!;

```

```

const userId = req.nextUrl.searchParams.get("userId");

await connectDB();

try {
  const data = await roomModel.findById(_id);
  // console.log("🔗 ~ file: route.ts:54 ~ data:", data)

  if(data === null) return NextResponse.json({
    error: "Not Found",
    path: "/player/room/invite"
  });

  if(data.members.find((user: any) => user.userId === userId) === null) {
    return NextResponse.json({
      error: "No Access",
      path: "/player/room/invite"
    });
  }

  const fetchedTracks = [];
  if(data.queue !== null){
    // if(data.queue.isUpdated){
    for(const trackId of data?.queue?.list){
      const track = await getSpotify('tracks', trackId.songId, null, {name: "market", value:
"ES"});
      fetchedTracks.push(track);
    }
    data.queue.isUpdated = false;
    data.save();
    // }
  }

  return NextResponse.json({
    queue: data.queue,
    fetchedTracks,
    _id: _id,
    isAdmin: data.adminId === userId,
    isActive: data.isActive,
    inviteCode: data.inviteCode,
    members: {
      count: data?.members.length
    }
  });
} catch (error) {
  console.log(error);
  return NextResponse.json(error);
}
}

```

Додаток Б
(Рекомендований)

КОПІЇ ПРЕЗЕНТАЦІЇ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки

Кваліфікаційна робота бакалавра на тему :

Розробка платформи для онлайн прослуховування музики

Виконав студент 4 курсу, групи ІТІР -20 -1
Семернін Максим Юрійович

Керівник:
Бітченко Олександр Миколайович

1

Мета роботи

Надати користувачам доступ до прослуховування музики безкоштовно, з додаванням функціоналу у вигляді чату, перекидання різних треків один одному, спільного прослуховування, перекладу текстів пісні. Інтерфейс та функціонал цього сайту має відповідати останнім стандартам, бути зручним як на комп'ютері, так і на телефоні

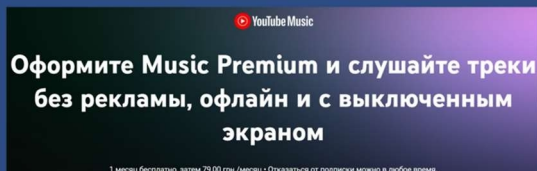
2

Задачі роботи

- провести огляд альтернативних сервісів;
- провести огляд технологій та API;
- розробити інтерфейс користувача;
- розробити відповідне програмне забезпечення.

3

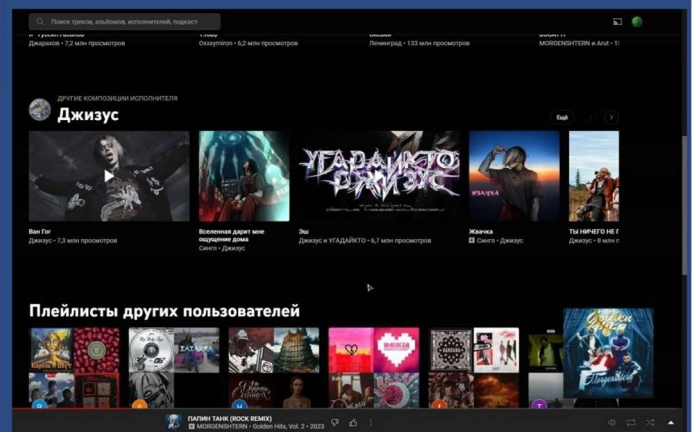
Youtube music



Скріншот реклами платної підписки



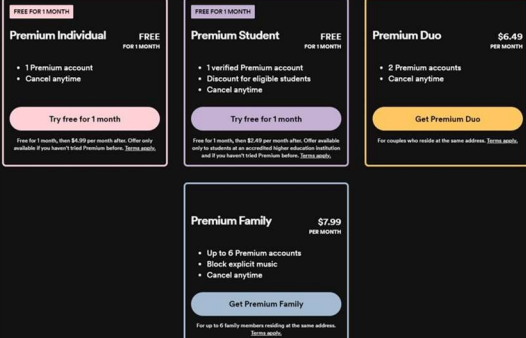
Скріншот реклами Youtube music




Скріншот рекомендованих плейлистів та артистів

4

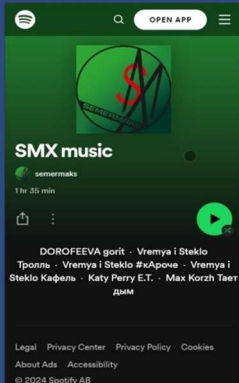
Spotify



Скріншот планів підписки



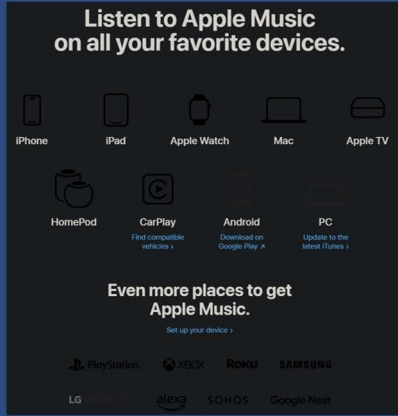
Реклама Spotify



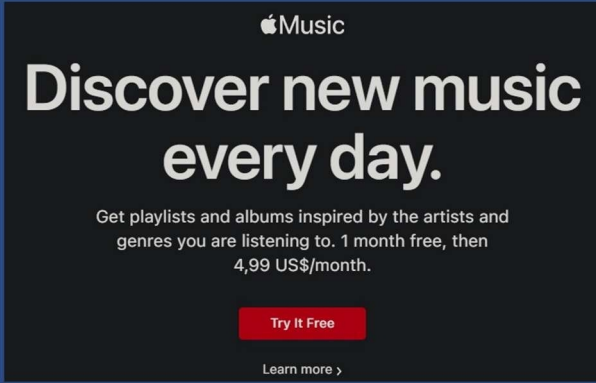
Скріншот мобільної версії інтерфейсу плейлиста

5

Apple music



Скріншот платформ для прослуховування на платній підписці



Скріншот реклами платної підписки

6

Технології та API

Logos displayed include: NEXT.JS, DeepL, Spotify API, mongoDB + node JS, AXIOS, and a cartoon bear character holding a sign that says "ZUSTAND".

The screenshot shows two versions of the M MUSIC website. The top version is in English, and the bottom version is in Ukrainian. A double-headed arrow indicates the relationship between the two versions.

English Version:

- The site was made by Ukrainian developer
- The best app for listening to music here and now.
- Listen to songs with your friends, share your favorite tracks in chat, all for free.
- No pauses, banners or ads.

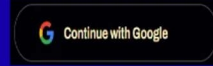
Ukrainian Version:

- Сайт зроблений Українським розробником
- Найкращий додаток для прослуховування музики тут і зараз.
- Слухайте пісні з друзями, поширюйте улюблені треки в чаті, і все це безкоштовно.
- Без пауз, банерів та реклами.

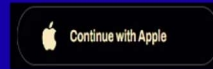
Реєстрація



NEXT - AUTH



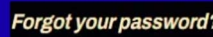
- Реєстрація за допомогою Google
акаунту



- Реєстрація за допомогою Apple
акаунту



- Реєстрація за допомогою акаунту
сайту



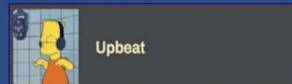
- Функція відновлювання паролю



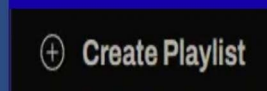
- Функція запам'ятовування паролю та
способу входу

9

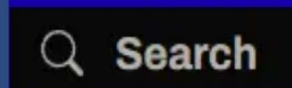
Головна сторінка



- Інтерфейс згенерованого плейліста



- Функція переходу до створювання
плейліста



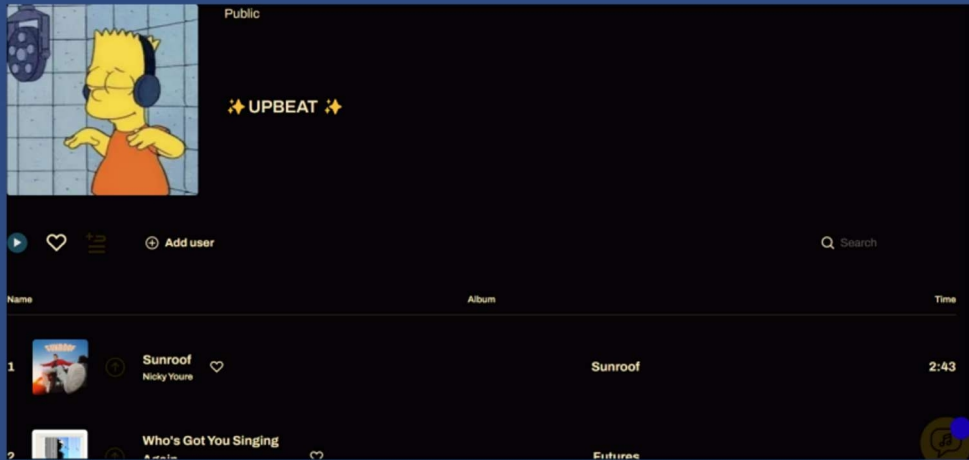
- Функція пошуку пісень



- Перехід до налаштувань

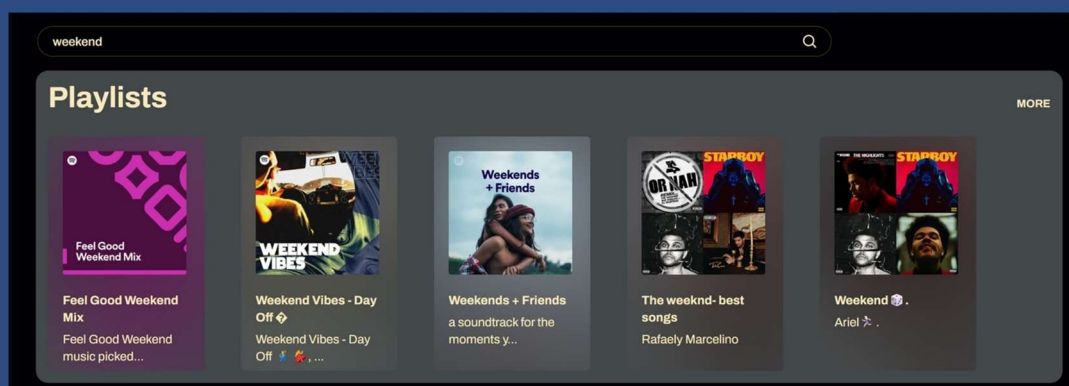
10

Плейліст



11

Пошук



12

Профіль та налаштування

Profile

Email

Username

Date of birth

Region

Change profile

- Зміна інформації
акаунту

Old password

Password

Forgot your password? Send me a reset mail

New password

New password

New password again










Change password

- Зміна паролю

13

Плеєр



-  - Рандомний мікс черги
-  - Попередня пісня
-  - Пауза/Продовжити
-  - Наступна пісня
-  - Повторювати пісню
-  - Налаштування гучності
-  - Додати пісню в плейліст
-  - Додати пісню улюбленню
-  - Інформація про пісню та автора

14

Чат

- Функція додавання друга

lanayyya
23 seconds ago

- Інформація про друга

- Функція сумісного прослуховування пісні

lanayyya invited you to a group listening!

- Інтерфейс сумісного прослуховування пісні

- Функція поширювання пісні

You shared a song with lanayyya!

- Інтерфейс поширювання пісні

IN 67108
від 07.06.2024

СЕРТИФІКАТ

учасника конференції

СЕМЕРНІН МАКСИМ ЮРІЙОВИЧ

Взяв(-ла) участь у VI Міжнародній студентській науковій конференції

ГЛОБАЛІЗАЦІЯ НАУКОВИХ ЗНАТЬ: МІЖНАРОДНА СПІВПРАЦЯ ТА ІНТЕГРАЦІЯ ГАЛУЗЕЙ НАУК

7 червня 2024 рік • м. Біла Церква, Україна

В рамках участі було опубліковано тези доповіді учасника на тему:

ПЛАТФОРМА ДЛЯ ОНЛАЙН ПРОСЛУХОВУВАННЯ МУЗИКИ

ДИРЕКТОР МОЛОДІЖНОЇ НАУКОВОЇ ЛІГИ
ГОЛОВА ОРГКОМІТЕТУ КОНФЕРЕНЦІЇ
ІГОР КОРЕНЮК

17

Додаток В
(Обов'язковий)

ВІДОМІСТЬ АТЕСТАЦІЙНОГО ПРОЕКТУ

