

Academia de Studii Economice din București
Facultatea de Cibernetică, Statistică și Informatică Economică

Proiect Cloud Computing - Documentație
NoteBuddy App

Profesor coordonator

Conf. Dr. Carmen Timofte

Student:

BALAN Theodor-Octavian

SIMPRE, Grupa 1131

Link-uri:

Link acces GitHub: <https://github.com/Cioger/CloudProject>

Link acces aplicație: <https://cloud-project-sooty.vercel.app/>

Link acces video prezentare: <https://youtu.be/Q6d1B4NuD90>

1. Introducere

În cadrul proiectului de seminar, am dezvoltat o aplicație pe nume "NoteBuddy", memorator personal. Scopul aplicației NoteBuddy este de a salva notite, cu titlu și categorie, pentru utilizator. Aceasta dispune de stocarea notitelor în baza de date și de un asistent personal sub forma unui model AI care poate furniza utilizatorului informații utile, care pot fi salvate în notes.

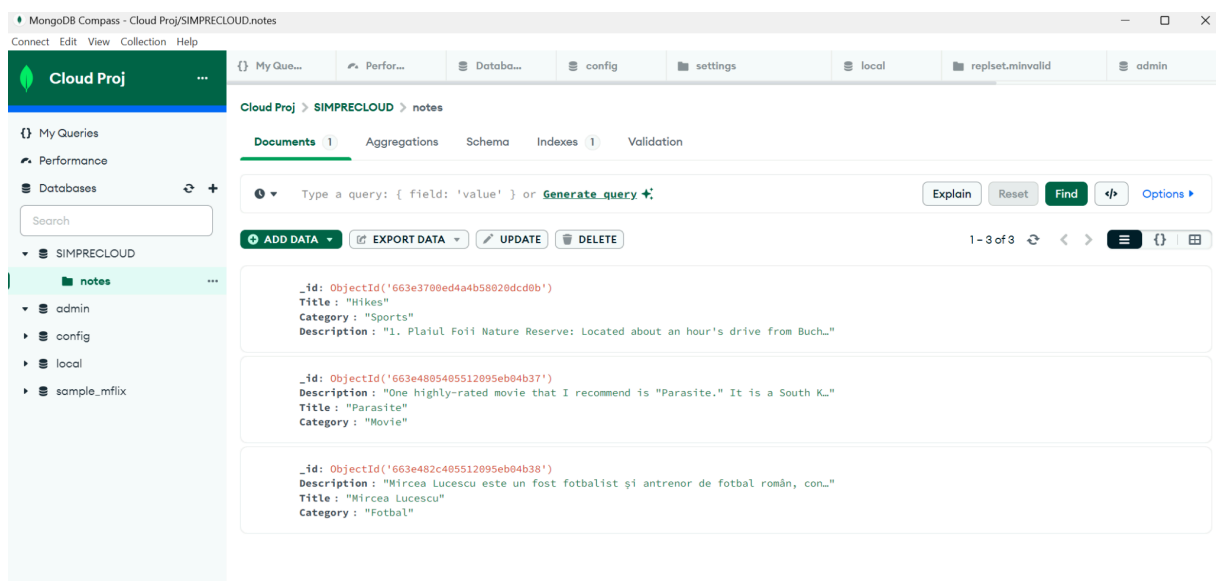
2. Descrierea problemei

Utilizatorii ar putea folosi aplicația "NoteBuddy" pentru a ține minte anumite detalii din viața de zi cu zi, sau pentru a nota răspunsurile diverselor curiozități pe care le pot avea. Câteva exemple de probleme pentru care aplicația ar putea oferi soluții sunt:

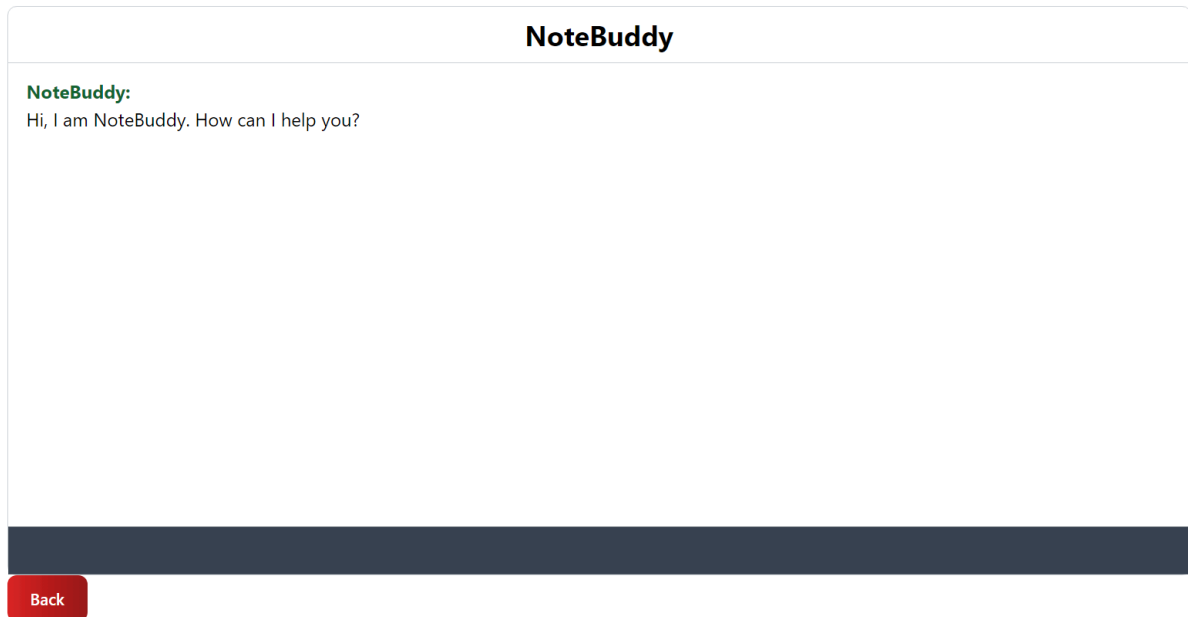
- ★ Planificare: Utilizatorii pot ține minte anumite evenimente care urmează să se desfășoare prin a le nota în aplicație.
- ★ Stocare informații: Utilizatorii pot nota informații de care au nevoie mai târziu.
- ★ Aflarea și stocarea răspunsurilor pentru curiozitățile cotidiene: Asistentul virtual poate oferi informațiile necesare pentru rezolvarea problemelor pe care utilizatorul le întâlnește, sau pur și simplu detalii despre subiecte asupra cărora cunoștințele utilizatorului sunt precare.

3. Descriere API

a) Am folosit **MongoDB** pentru salvarea datelor introduse în cadrul aplicației:



b) API-ul de **ChatGPT** ajută aplicația să devină o punte de legătură cu o sursă infinită de informații de actualitate.



4. Fluxul de date

Homepage:

```
pages > JS index.js > ...
1  import MainPage from "@components/MainPage";
2
3  export default function Home() {
4    return (
5      <MainPage />
6    );
7  }
8
```

```

components > MainPage.jsx > [1] MainPage > data.map() callback
6   const MainPage = () => {
7
8   }
9
10  return (
11    <section className="bg-white dark:bg-blue-100">
12      <div className="container px-6 py-10 mx-auto">
13        <h1 className="w-[500px] mx-auto text-center text-blue-800 text-6xl">NoteBuddy</h1>
14        <div className="p-4 flex flex-wrap gap-4">
15          {data?.map?.(note) => (
16            <div
17              key={note._id}
18              className="max-w-sm p-6 bg-white border border-gray-200 rounded-lg shadow dark:bg-blue-500 dark:border-blue-700"
19            >
20              <h5 className="mb-2 text-2xl font-bold tracking-tight text-gray-900 dark:text-white">
21                {note.Title}
22              </h5>
23              <h5 className="mb-2 text-2xl font-bold tracking-tight text-gray-900 dark:text-white">
24                Category: {note.Category}
25              </h5>
26              <p className="mb-3 font-normal text-white dark:text-text-white">
27                {note.Description}
28              </p>
29              <button
30                type="button"
31                onClick={() => handleEditNote(note._id)}
32                className="text-white bg-gradient-to-r from-green-600 via-green-700 to-green-800 hover:bg-gradient-to-br focus:ring-4 focus:outline-none focus:ring-green-300 dark:focus:ring-4"
33              >
34                Update
35              </button>
36              <button
37                type="button"
38                onClick={() => handleDeleteNote(note._id)}
39                className="text-white bg-gradient-to-r from-red-600 via-red-700 to-red-800 hover:bg-gradient-to-br focus:ring-4 focus:outline-none focus:ring-red-300 dark:focus:ring-4"
40              >
41                Delete
42              </button>
43            </div>
44          )}
45        </div>
46      </div>
47    </section>
48  )
49
50  }
51
52  }
53
54  }
55
56  }
57
58  }
59
60  }
61
62  }
63
64  }
65
66  }
67
68  }
69
70  }
71
72  }
73
74  }
75
76  }
77
78  }
79
80  }
81
82  }
83
84  }
85
86  }
87
88  }
89
90  }
91
92  }
93
94  }
95
96  }
97
98  }
99
100 }

```

ChatGPT API:

```

components > ChatComponent.jsx > [1] ChatComponent
1   import { useState } from "react";
2   import MessageBox from "../MessageBox";
3   import { askChat } from "@utils/chatFunctions";
4   import { useRouter } from "next/router";
5
6   const ChatComponent = () => {
7     const router = useRouter();
8     const [chatMessages, setChatMessages] = useState([]);
9     const [isLoading, setIsLoading] = useState(false);
10    const [userInput, setUserInput] = useState("");
11
12    const filterChatHistory = (chatHistory) => {
13      let filteredChatHistory = [];
14
15      for (let i = 0; i < chatHistory.length; i++) {
16        const currMessage = chatHistory[i];
17        const nextMessage = chatHistory[i + 1];
18
19        if (i == chatHistory.length - 1 || (currMessage.type !== 'error' && nextMessage?.type !== 'error' && currMessage.role !== nextMessage?.role)) {
20          filteredChatHistory.push(currMessage);
21        }
22      }
23
24      return filteredChatHistory;
25    }
26
27    const handleBack = () => {
28      router.push("/");
29    }
30
31    const handleKeyDown = async (e) => {
32      if (e.key !== 'Enter' || userInput.length === 0) return;
33
34      const currentMessages = {
35        role: 'user',

```

```

pages > chat.jsx > ...
1   import ChatComponent from '@components/ChatComponent'
2   import React from 'react'
3
4   const Chat = () => {
5     return (
6       <ChatComponent />
7     )
8   }
9
10  export default Chat

```

```

pages > api > answer.js > ...
1  import openai from "@lib/openai";
2  import { sendBadRequest, sendOk } from "@utils/apiMethods";
3  import { MAX_MEMORY } from "@utils/constants";
4
5  const SYSTEM_PROMPTS = {
6    SIMPLE_ASSISTANT : {
7      MESSAGE: {
8        'role': 'system',
9        'content': 'You are a simple assistant. You respond with simple sentences.'
10     },
11     TEMPERATURE : 1,
12     MAX_TOKENS: 50,
13     TYPE: 'simple_assistant'
14   },
15   USER: {
16     MESSAGE: {
17       'role': 'user',
18       'content': 'You are a user. You are asking the assistant for help.'
19     },
20     TEMPERATURE : 1,
21     MAX_TOKENS: 1000,
22     TYPE: 'user'
23   }
24 }
25
26 const chatCompletion = async (messages, max_tokens, temperature) => {
27   const rawResponse = await openai.chat.completions.create({
28     model: 'gpt-3.5-turbo',
29     messages: messages,
30     max_tokens: max_tokens,
31     temperature: temperature
32   });

```

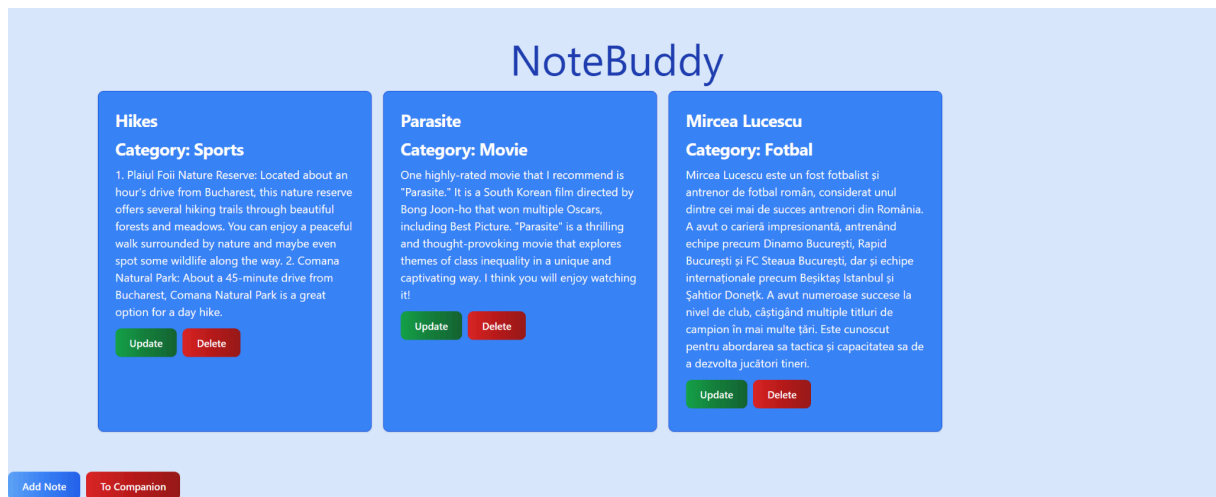
Formular Add/Update:

```

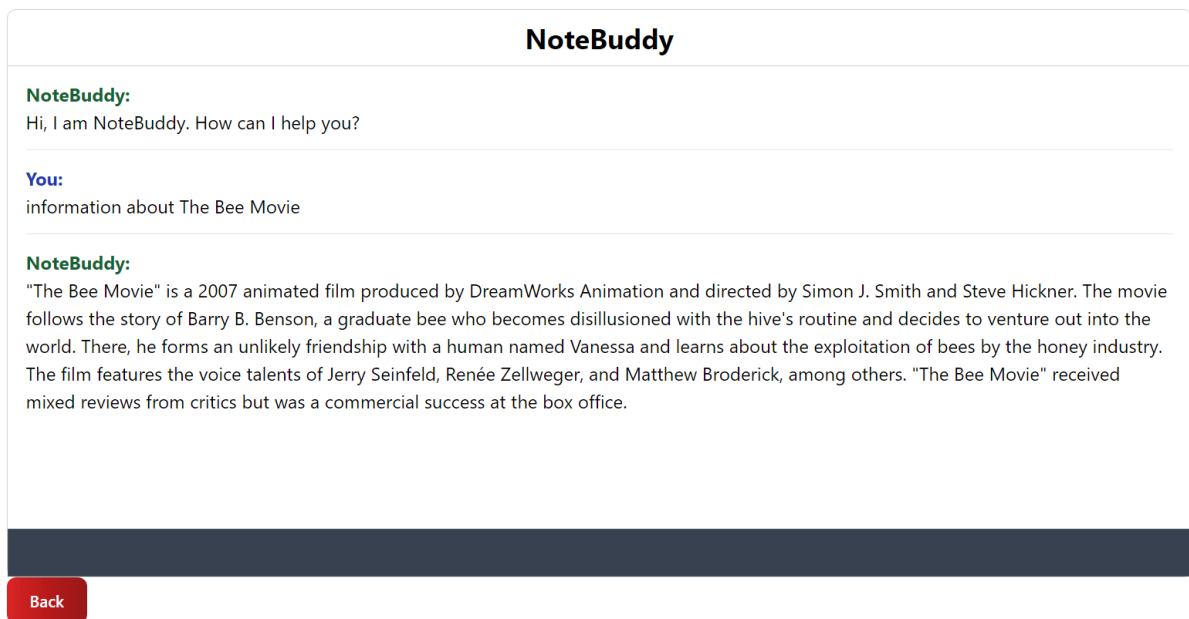
components > NoteForm.jsx > NoteForm
4  const NoteForm = (props) => {
5    const router = useRouter();
6    const { entry, onSubmit } = props;
7    const [data, setData] = useState(entry);
8
9    const handleChange = (type, value) => {
10      setData({ ...data, [type]: value });
11    };
12
13    const handleCancel = () => {
14      router.push("/");
15    }
16
17    return (
18      <div className="p-4">
19        <div className="flex flex-col mx-auto max-w-80 border p-4 shadow-sm gap-4 w-full">
20          <div className="text-center font-bold text-xl">{entry._id ? 'Update' : 'Create new'} Notes</div>
21          <div>
22            <label>
23              <htmlFor="Title"
24                className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"
25            >
26              Title
27            </label>
28            <input
29              type="Title"
30              id="Title"
31              value={data.Title}
32              onChange={(e) => handleChange("Title", e.target.value)}
33              className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-
34              placeholder="Title placeholder"
35              required
36            />
37          </div>
38        </div>
39      </div>

```

5. Capturi de ecran ale aplicației



HomePage



Exemplu de utilizare pagina ChatGPT

Create new Note

Title placeholder

Category placeholder

Write your thoughts here...

Cancel

Create

Pagina de Create – Exemplu de create obiect

NoteBuddy

Hikes

Category: Sports

1. Plaiul Foii Nature Reserve: Located about an hour's drive from Bucharest, this nature reserve offers several hiking trails through beautiful forests and meadows. You can enjoy a peaceful walk surrounded by nature and maybe even spot some wildlife along the way. 2. Comana Natural Park: About a 45-minute drive from Bucharest, Comana Natural Park is a great option for a day hike.

Update

Delete

Parasite

Category: Movie

One highly-rated movie that I recommend is "Parasite." It is a South Korean film directed by Bong Joon-ho that won multiple Oscars, including Best Picture. "Parasite" is a thrilling and thought-provoking movie that explores themes of class inequality in a unique and captivating way. I think you will enjoy watching it!

Update

Delete

Mircea Lucescu

Category: Fotbal

Mircea Lucescu este un fost fotbalist și antrenor de fotbal român, considerat unul dintre cei mai de succes antrenori din România. A avut o carieră impresionantă, antrenând echipe precum Dinamo București, Rapid București și FC Steaua București, dar și echipe internaționale precum Beşiktaş Istanbul și Șahtior Donețk. A avut numeroase succese la nivel de club, câștigând multiple titluri de campion în mai multe țări. Este cunoscut pentru abordarea sa tactică și capacitatea sa de a dezvolta jucători tineri.

Update

Delete

Joke to remember

Category: Entertainment

Why couldn't the bicycle stand up by itself? Because it was two tired!

Update

Delete

HomePage – cu noul note adăugat

6. Referințe

Curs Cloud Computing - <https://online.ase.ro/course/view.php?id=40607>

Seminar Cloud Computing - <https://online.ase.ro/course/view.php?id=40613>