**מפרט ארכיטקטוני של המערכת**

**ניהול קורונה לקופת חולים**

שכבת הנתונים

מאגר המידע בmongo DB.

שם הdata base : coronaReservior

שם ה: schema Clients.

Schema Clients - מתארת עבור כל חבר בקופת חולים את פרטיו האישים ובנושא הקורונה.

פרטי לקוח:

 fullName : String, required

  idCard : String, required

  address : String, required

  dateOfBirth :Date, required

  mobilePhone : String, required

  telephone : String, required

  positive\_result\_date :Date

  recovery\_date :Date

  vaccination\_dates :[ Date ]

  vaccine\_manufacturers :[ String ]

שכבת היישום

צד השרת NODE.JS -

שמכיל:

1. תיקיית models- מכילה את Schema Clients
2. תיקיית controllers – מכילה פונקציות:addClient getAllClients
3. תיקיית routers- מכילה ניתובים לפונקציות ב controllers

שכבת המצגת

צד לקוח, API –REACT

שכולל את :

1. App.js – דף ראשי. מכיל את הקומפוננטה Routings
2. Index.js – הדף שרץ בדפדפן. רינדור הקומפוננטה App(פעם אחת).
3. Api.js – getAllClients : קבלת כל מערך הלקוחות מהמסד נתונים. addClient: הוספת לקוח למסד נתונים.
4. Routings.js – מכיל את הניתובים לדפים השונים.
5. Layout.js – מערך, סרגל הניתובים ל:

- Sign In : כניסת לקוח

- :Sign Up הרשמת לקוח

- Manager: כפתור לגישת מנהל

1. SignUp.js – הרשמה. הוספת לקוח למסד נתונים.
2. SignIn.js – כניסה לפרטי הלקוח.
3. Manager.js – כניסה למנהל קופת חולים ע"י הקשת קוד(1234), ואפשור גישה לדפים הבאים:
4. Clientsdetails.js – פרטי כל הלקוחות.
5. Activeclients.js – חישוב מספר לקוחות חולים בכל יום לפי חודש קודם.
6. Unvacciatedclients.js – חישוב מספר לקוחות לא מחוסנים.

**החיבור בין השכבות**

המסד נתונים לשרת:

קובץ .env

DB\_CONNECT="mongodb+srv://shira:shira214124307@cluster0.mp7fcd1.mongodb.net/coronaReservoir?retryWrites=true&w=majority"

PORT=8000

בקובץ app.js

mongoose.connect(process.env.DB\_CONNECT, connectionParams)

    .then(() => {

        console.log('connected to db')

    }).catch(err => {

        console.log(err)

    })

app.listen(process.env.PORT, () => { console.log(`listening port ${process.env.PORT}`) })

השרת לצד לקוח:

הגדרה בAPI נקודות קצה עבור פעולות שונות, כגון קבלת נתונים, יצירת נתונים, עדכון נתונים ומחיקת נתונים.

getAllClients – הצגת כל הלקוחות מהמסד נתונים

export async function getAllClients() {

    try {

          const res = await fetch('http://localhost:8000/clients', { method: 'GET' });

          const data = await res.json();

          return data;

      } catch (err) {

          console.log(err);

      }

}

addClient- הוספת לקוח למסד נתונים

  export async function addClient(newPost) {

    const myHeaders = new Headers();

    myHeaders.append("Content-Type", "application/json");

    const requestOptions = {

        method: 'POST',

        headers: myHeaders,

        body: JSON.stringify(newPost),

    };

    try {

          const res = await fetch('http://localhost:8000/clients', requestOptions);

          const data = await res.json();

          return data;

      } catch (err) {

          console.log(err);

      }

}

או באמצעות :axios

  useEffect(() => {

    if (clients.length === 0)  {

      axios.get('http://localhost:8000/clients/')

        .then(response => {

          setClients(response.data);

        })

        .catch(err => console.log(err));

    }

  }, [clients])