תיק פרויקט
יובל בן-שחר
יולי 2025

1)
אפיון ותכנון הפרויקט

שם ותיאור הפרויקט

שם הפרויקט: "בנו מחשב!"

תיאור כללי: אפליקציית ה-web מאפשרת למשתמש לבחור חלקי מחשב עבור בנייה של אחד. האפליקציה מחשבת את צריכת החשמל ועלות הבניה כדי לתת למשתמש מידע כמה שיותר מדויק על המחשב המתוכנן.

בעיה שהפרויקט פותר: אנשים מעוניינים לבנות מחשב בעצמם אבל נבהלים מתהליך בחירת החלקים שעלול להיתפס כמסורבל. המערכת מספקת לאותם אנשים כלי פשוט וידידותי, בעזרתו יוכלו לתכנן בנייה של מחשב.

קהל יעד: כל אדם המעוניין לבנות מחשב בעצמו.

דרישות מערכת

דרישות פונקציונליות:

- המשתמש יכול להירשם ולהתחבר למערכת.
- המשתמש יכול לבחור ולהסיר חלקים מרשימה.
- המערכת מציגה למשתמש את עלות ההרכבה וצריכת החשמל.
- המשתמש יכול לשמור בניות של מחשב והמערכת תציג את הבניה האחרונה בעת התחברות.

דרישות טכניות:

- Frontend: HTML, CSS, JavaScript
- Backend: Node.JS, Express
- Database: SQLite3

2)
תכנון מבנה האפליקציה

מבנה תיקיית הפרויקט

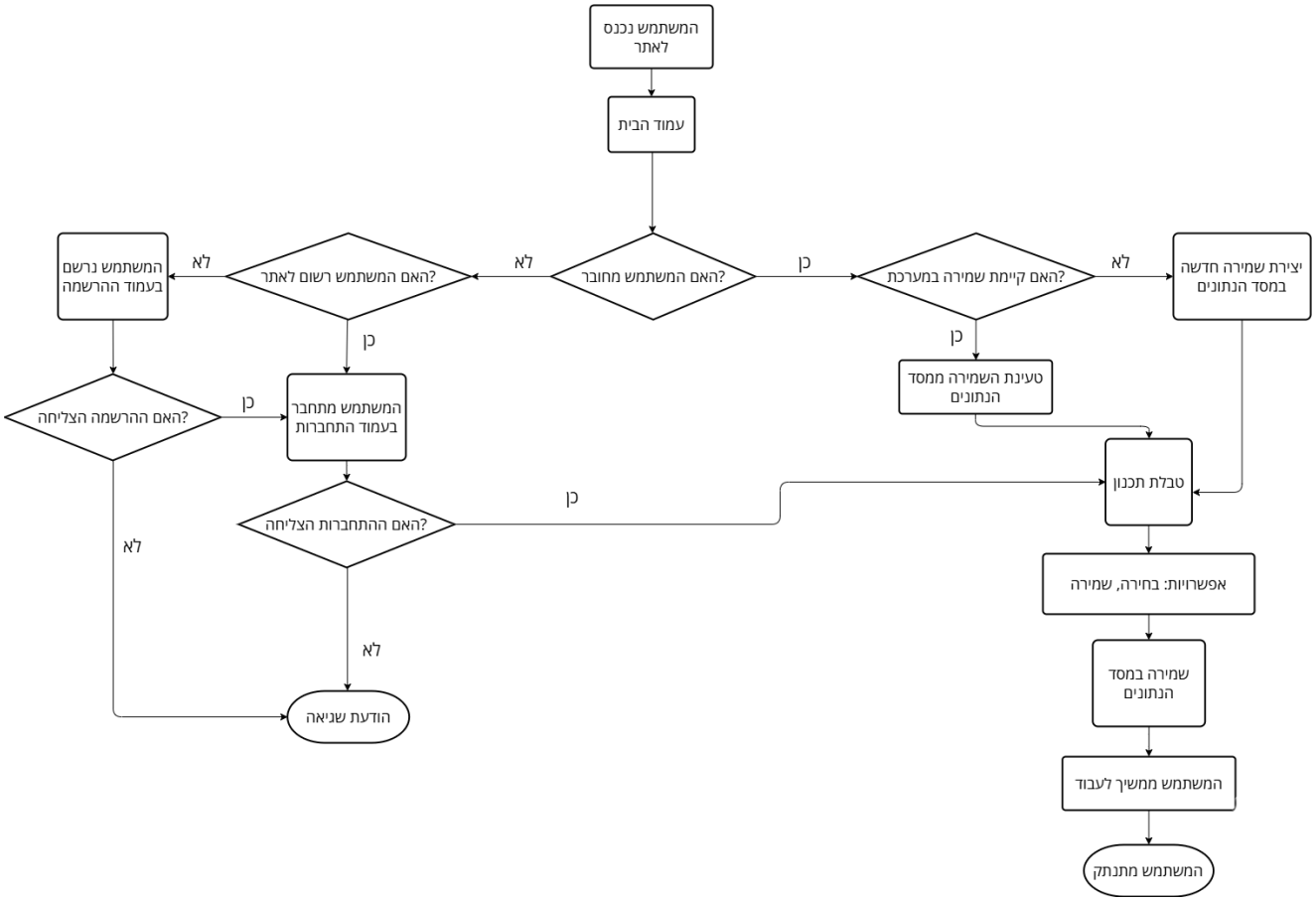
```
pcbuilder/
    ├── package.json
    ├── public/
    ├── style/
    ├── backend/
    │   └── database/
    └── src/
└── img/
```

- /public - מכיל קבצי html.
- /backend - מכיל את קוד השרת.
- /database - מכיל את מסדי הנתונים ודואג לחיבורם.
- /style - מכיל את קבצי העיצוב.
- /src - מכיל תמונות ומדיה.

תרשים זרימה של הניווט

מסכים עיקריים:

- עמוד הבית - מציג את פעולות האפליקציה ומתפקד כעמוד כניסה.
- עמוד "אודות" - מסביר אודות האתר ומטרותיו.
- עמוד "יצירת קשר" - מאפשר שליחת טופס יצירת קשר למפתח האתר.
- עמוד התחברות - מאפשר את התחברות המשתמשים לחשבונם במערכת.
- עמוד הרשמה - מאפשר הרשמה של משתמשים במערכת.
- עמוד העריכה (dashboard) - מאפשר תקשורת עם המערכת עצמה ועריכה של מחשב בטבלת תכנון. לעמוד זה ניתן לגשת רק לאחר התחברות.


המשתמש נכנס לאתר
עמוד הבית
?האם המשתמש מחובר
כן
לא
?האם קיימת שמירה במערכת
לא
יצירת שמירה חדשה במסד הנתונים
כן
טעינת השמירה ממסד הנתונים
?האם המשתמש רשום לאתר
לא
המשתמש נרשם בעמוד ההרשמה
כן
המשתמש מתחבר בעמוד התחברות
?האם ההרשמה הצליחה
כן
לא
הודעת שגיאה
?האם ההתחברות הצליחה
כן
לא
טבלת תכנון
אפשרויות: בחירה, שמירה
שמירה במסד הנתונים
המשתמש ממשיך לעבוד
המשתמש מתנתק

3)

פיתוח צד הלקוח

- index.html - מציג את פעולות האפליקציה ומתפקד כעמוד כניסה.
- about.html - מסביר אודות האתר ומטרותיו.
- contact.html - מאפשר שליחת טופס יצירת קשר למפתח האתר.
- login.html - מאפשר את התחברות המשתמשים לחשבונם במערכת.
- register.html - מאפשר הרשמה של משתמשים במערכת.
- main.html - מאפשר תקשורת עם המערכת עצמה ועריכה של מחשב בטבלת תכנון. לעמוד זה ניתן לגשת רק לאחר התחברות.

קטע קוד לדף התחברות (login.html):

```
<form action="/login" method="post">
  <div class="login_input_field">
    <input type="email" name="email" placeholder="אי-מייל" required />
  </div>
  <div class="login_input_field">
    <input type="password" name="password" placeholder="סיסמה" required />
  </div>
  <div id="remember_me_register">
    <div class="align_rmr">
      <input type="checkbox" id="remember_me_btn" name="remember" />
      <label for="remember_me_btn">זכור אותי</label>
    </div>
    <div class="align_rmr">
      <a href="register" id="register_in_login">הרשמה</a>
    </div>
  </div>
  <div id="button_align">
    <button class="enter_button" type="submit">כניסה</button>
  </div>
</form>
```

- <form> - מגדירה טופס לאיסוף נתונים מהמשתמש עבור השרת.
- <”input type=”email> - שדה קלט עבור המייל.
- <”input type=”password> - שדה קלט עבור הסיסמה, הטקסט מתכסה בנקודות.
- <”input type=”checkbox> - תיבה לסימון רצון להיזכר על ידי המערכת.
- required - שדה חובה, מונע השארת שדות קלט ריקים.
- placeholder - טקסט בתוך תיבות הקלט שמסמן למשתמש היכן להכניס את פרטי החשבון.

- - שיוך לקבוצה, השימוש נעשה בעיקר לצרכים עיצוביים.
- <label> - תווית לשדה לתיבת הסימון, למען ההבנה של המשתמש.
- <a href="register"> - קישור לעמוד ההרשמה.
- <button type="submit"> - כפתור לשליחת הטופס לשרת.

<u>קוד צד הלקוח:</u>

<u>קוד להתחברות:</u>

```
app.post("/login", (req, res) => {
  const { email, password, remember } = req.body;

  database.get(
    `SELECT * FROM users WHERE email = ? AND password = ?`,
    [email, password],
    (err, row) => {
      if (err) {
        console.error(err);
        return res.status(500).send("שגיאה בהתחברות");
      }

      if (!row) {
        return res.status(401).send("פרטי התחברות שגויים");
      }

      req.session.user = { id: row.id, email: row.email };

      if (remember) {
        req.session.cookie.maxAge = 7 * 24 * 60 * 60 * 1000;
      } else {
        req.session.cookie.expires = false;
      }

      return res.redirect("/dashboard");
    }
  );
});
```

יצירת משתנה

שאילת מידע ממסד הנתונים

פונקציות להודעה במקרה של שגיאה

התחברות לסשן

הורדת עוגיות לזכירת המשתמש במידה והוא מעוניין בכך

שליחה לעמוד טבלת העריכה

קוד לשמירת בנייה:

```
app.post("/save-build", (req, res) => {
  const { email, build } = req.body;

  if (!email || !build) {
    return res.status(400).json({ error: "Missing email or build data" });
  }

  const buildJson = JSON.stringify(build);

  buildDB.run(
    `DELETE FROM build WHERE email = ?`,
    [email],
    function (deleteErr) {
      if (deleteErr) {
        console.error(deleteErr);
        return res.status(500).json({ error: "Error deleting previous builds" });
      }

      buildDB.run(
        `INSERT INTO build (email, build_data) VALUES (?, ?)`,
        [email, buildJson],
        function (insertErr) {
          if (insertErr) {
            console.error(insertErr);
            return res.status(500).json({ error: "Error saving new build" });
          }

          res.status(200).json({ message: "Build saved successfully", buildId: this.lastID });
        }
      );
    }
  );
});
```

יצירת משתנה עבור ערכי המייל והבניה

הודעת שגיאה במקרה והנתונים חסרים

יצירת משתנה חדש המאחסן את נתוני הבנייה,
הפיכת הנתונים לטקסט מJSON

מחיקת הבנייה הקודמת ממסד הנתונים

הודעת שגיאה במקרה ודברים לא הולכים כראוי

הזנת המידע החדש בתוך מסד הנתונים

הודעת שגיאה במקרה ודברים לא הולכים כראוי

הודעת הצלחה למשתמש

קוד הטבלה:

```
document.addEventListener("DOMContentLoaded", async () => {
  const partTypes = ["cpu", "gpu", "motherboard", "cooling", "psu", "storage", "case"];
  const emailDisplay = document.getElementById("email_display");
  let parts = [];

  try {
    const [partsRes, sessionRes] = await Promise.all([
      fetch("/parts"),
      fetch("/session-info")
    ]);

    if (!partsRes.ok || !sessionRes.ok) throw new Error("Failed to load data");

    parts = await partsRes.json();
    const { email } = await sessionRes.json();
    emailDisplay.textContent = email;

    const calculateTotals = () => {
      let totalPrice = 0;
      let totalWatt = 0;

      partTypes.forEach(type => {
        const select = document.getElementById(`${type}_select`);
        const selectedOption = select.selectedOptions[0];
        if (selectedOption && selectedOption.value !== "") {
          totalPrice += parseInt(selectedOption.dataset.price);
          totalWatt += parseInt(selectedOption.dataset.watt);
        }
      });

      document.getElementById("total-price").textContent = "מחיר כולל: ₪" + totalPrice;
      document.getElementById("total-wattage").textContent = "צריכת חשמל: " + totalWatt + "W";
    };

    const populateSelects = () => {
      partTypes.forEach(type => {
        const select = document.getElementById(`${type}_select`);
        select.setAttribute("data-part-type", type);

        const relevantParts = parts.filter(p => p.type === type);

        const defaultOption = document.createElement("option");
        defaultOption.textContent = `בחר ${type}`;
        defaultOption.disabled = true;
        defaultOption.selected = true;
        defaultOption.value = "";
        select.appendChild(defaultOption);

        relevantParts.forEach(part => {
          const option = document.createElement("option");
          option.value = part.id;
          option.textContent = `${part.name} - ₪${part.price} - ${part.watt}W`;
          option.dataset.price = part.price;
          option.dataset.watt = part.watt;
          select.appendChild(option);
        });

        select.addEventListener("change", () => {
          const selectedOption = select.selectedOptions[0];
          const wattCell = select.parentElement.nextElementSibling;
          const priceCell = wattCell.nextElementSibling;

          wattCell.textContent = selectedOption.dataset.watt + "W";
          priceCell.textContent = "₪" + selectedOption.dataset.price;

          calculateTotals();
        });
      });
    };

    populateSelects();

    fetch(`/last-build?email=${encodeURIComponent(email)}`)
      .then(res => res.json())
      .then(savedBuild => {
        if (Array.isArray(savedBuild)) {
          savedBuild.forEach(item => {
            const select = document.getElementById(`${item.partType}_select`);
            if (!select) return;
            const option = Array.from(select.options).find(opt => opt.value == item.partId);
            if (option) {
              option.selected = true;
              const wattCell = select.parentElement.nextElementSibling;
              const priceCell = wattCell.nextElementSibling;
              wattCell.textContent = option.dataset.watt + "W";
              priceCell.textContent = "₪" + option.dataset.price;
            }
          });
          calculateTotals();
        }
      });

    document.getElementById("saveBuildBtn").addEventListener("click", () => {
      const build = [];

      document.querySelectorAll(".select-cell select").forEach(select => {
        const selected = select.options[select.selectedIndex];
        if (selected && selected.value) {
          build.push({
            partType: select.getAttribute("data-part-type") || "unknown",
            partName: selected.textContent,
            partId: selected.value
          });
        }
      });

      if (build.length === 0) {
        alert("לא נבחרו רכיבים לשמירה");
        return;
      }

      fetch("/save-build", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ email, build })
      })
        .then(res => res.json())
        .then(data => {
          if (data.message) {
            alert("המערכת נשמרה בהצלחה!");
          } else {
            alert("אירעה שגיאה בשמירת המערכת.");
          }
        })
        .catch(err => {
          console.error("Error saving build:", err);
          alert("אירעה שגיאה.");
        });
    });

  } catch (err) {
    console.error("Initialization error:", err);
    window.location.href = "/login";
  }
});
```

קוד צד השרת:

קטע מקוד השרת:

```
const express = require("express");
const session = require("express-session");
const path = require("path");
const sqlite3 = require("sqlite3");
const database = require("./database/database");
const partsDB = require("./database/partsDatabase");
const buildDB = require("./database/buildDatabase");

const app = express();

app.use(session({
  secret: "super-secret-key",
  resave: false,
  saveUninitialized: false,
  cookie: {
  secure: process.env.NODE_ENV === "production",
  httpOnly: true,
  sameSite: "strict"
}
}));


app.use(express.static(path.join(__dirname, "../style")));
app.use(express.static(path.join(__dirname, "../public")));
app.use(express.static(path.join(__dirname, "../src/img")));
app.use(express.urlencoded({ extended: true }));
app.use(express.json());
```

חיוב השימוש בחבילות מותקנות

חיוב השימוש במסדי הנתונים

קביעת הגדרות הסשן בעת יצירת סשן חדש

טעינת קבצים ותמונות הקשורים בעיצוב

שימוש בחבילות

קוד מסד הנתונים

קוד מסד הנתונים של רישום משתמשים:

```
const sqlite3 = require("sqlite3");
const path = require("path");

const dbPath = path.join(__dirname, "users.db");
const database = new sqlite3.Database(dbPath);

database.serialize(() => {
  database.run(
    `CREATE TABLE IF NOT EXISTS users (
      id INTEGER PRIMARY KEY AUTOINCREMENT,
      email TEXT NOT NULL UNIQUE,
      password TEXT NOT NULL
    )`,
    (err) => {
      if (err) {
        console.error("Error creating users table:", err);
      } else {
        console.log("Users table ready.");
      }
    }
  );
});

module.exports = database;
```

קוד מסד הנתונים של החלקים:

```
const sqlite3 = require("sqlite3");
const path = require("path");

const partPath = path.join(__dirname, "parts.db");
const partsDB = new sqlite3.Database(partPath, (err) => {
  if (err) {
    console.error("Failed to connect to parts database", err);
  } else {
    console.log("Connected to parts database");

    partsDB.run(`
      CREATE TABLE IF NOT EXISTS parts (
        id INTEGER PRIMARY KEY,
        type TEXT NOT NULL,
        name TEXT NOT NULL,
        price INTEGER NOT NULL,
        watt INTEGER NOT NULL
      );
    `, () => {
      partsDB.get(`SELECT COUNT(*) AS count FROM parts`, (err, row) => {
        if (err) return console.error("Failed to check parts count", err);
        if (row.count > 0) return ;

        partsDB.run(`
          INSERT INTO parts (type, name, price, watt) VALUES
          -- CPUs
          ('cpu', 'Intel Core i5-12400', 850, 65),
          ('cpu', 'Intel Core i7-13700K', 1800, 125),
          ('cpu', 'AMD Ryzen 5 5600X', 780, 65),
          ('cpu', 'AMD Ryzen 7 5800X', 1250, 105),
          ('cpu', 'Intel Core i3-12100F', 450, 60),

          -- GPUs
          ('gpu', 'NVIDIA RTX 3060', 1800, 170),
          ('gpu', 'NVIDIA RTX 4060 Ti', 2200, 160),
          ('gpu', 'AMD RX 6600', 1400, 132),
          ('gpu', 'AMD RX 6700 XT', 2000, 230),
          ('gpu', 'NVIDIA GTX 1660 Super', 1000, 125),

          -- Motherboards
          ('motherboard', 'ASUS B550', 500, 50),
          ('motherboard', 'MSI B450 Tomahawk', 430, 45),
          ('motherboard', 'Gigabyte B660M DS3H', 480, 55),
          ('motherboard', 'ASUS TUF Gaming Z690', 880, 60),
          ('motherboard', 'ASRock B550M Steel Legend', 540, 50),

          -- Coolers
          ('cooling', 'Cooler Master Hyper 212', 150, 10),
          ('cooling', 'Noctua NH-D15', 420, 12),
          ('cooling', 'be quiet! Pure Rock 2', 210, 9),
          ('cooling', 'Corsair H100i RGB', 620, 15),
          ('cooling', 'Arctic Freezer 34', 180, 10),

          -- PSUs
          ('psu', 'Corsair 650W Bronze', 320, 0),
          ('psu', 'Seasonic 750W Gold', 520, 0),
          ('psu', 'EVGA 600W White', 280, 0),
          ('psu', 'Cooler Master 850W Gold', 580, 0),
          ('psu', 'Corsair RM1000x', 690, 0),

          -- Storage
          ('storage', 'Samsung 970 EVO 1TB', 420, 5),
          ('storage', 'WD Blue 1TB HDD', 180, 4),
          ('storage', 'Kingston NV2 500GB', 230, 3),
          ('storage', 'Seagate Barracuda 2TB', 260, 5),
          ('storage', 'Crucial MX500 1TB', 360, 4),

          -- Cases
          ('case', 'NZXT H510', 300, 0),
          ('case', 'Corsair 4000D Airflow', 390, 0),
          ('case', 'Fractal Design Meshify C', 420, 0),
          ('case', 'Cooler Master NR600', 350, 0),
          ('case', 'Lian Li Lancool 215', 370, 0);
        `, (err) => {
          if (err) console.error("Error inserting parts:", err);
          else console.log("Sample parts inserted into database.");
        });
      });
    });
  }
});

module.exports = partsDB;
```

קוד מסד הנתונים של שמירת הבניות:

```
const sqlite3 = require("sqlite3");
const path = require("path");

const buildDB = new sqlite3.Database(path.join(__dirname, "build.db"));

buildDB.serialize(() => {
  buildDB.run(`
    CREATE TABLE IF NOT EXISTS build (
      id INTEGER PRIMARY KEY AUTOINCREMENT,
      email TEXT NOT NULL,
      build_data TEXT NOT NULL,
      created_at DATETIME DEFAULT CURRENT_TIMESTAMP
    )
  `);
});

module.exports = buildDB;
```

מצגת:

יובל בן-שחר, יולי 2025

תיאור הפרויקט

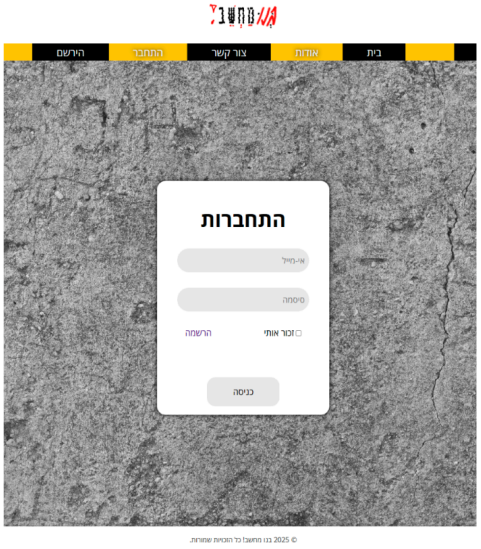
הפרויקט הוא אפליקציה של טבלת עריכה המאפשרת למשתמש לתכנן בעזרתה בנייה של מחשב.

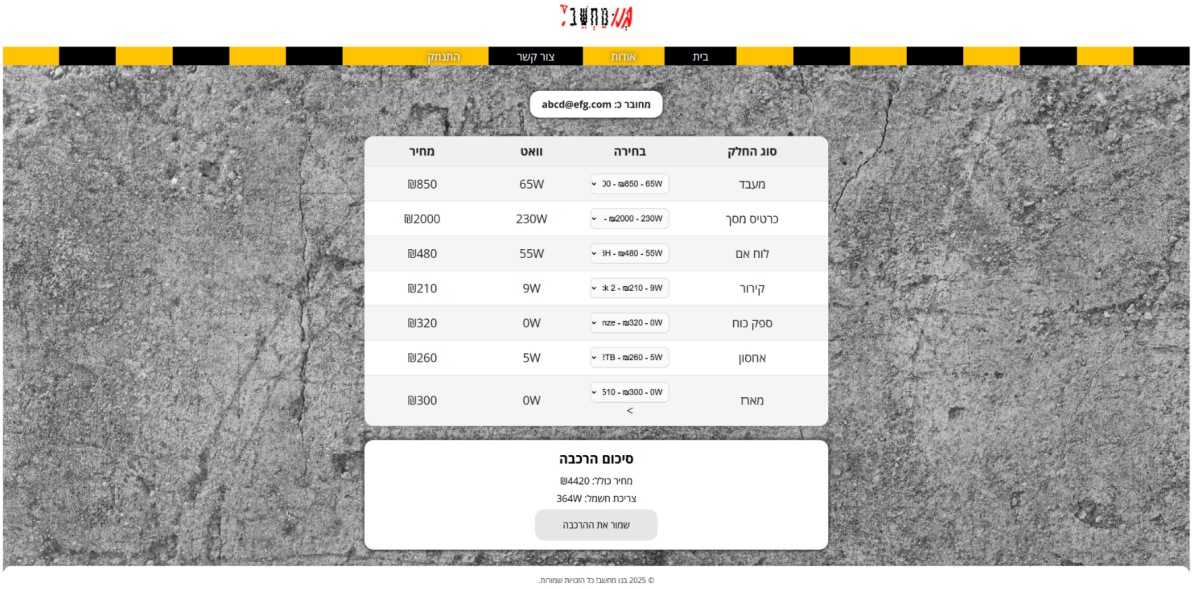
המשתמשים יכולים להירשם, להתחבר, לבחור חלקים ולשמור במערכת את אשר עשו.

המערכת טוענת באופן אוטומטי את הבניה האחרונה בעת ההתחברות.

צילומי מסך


בית
אודות
צור קשר
התחבר
הירשם
תכננו את מחשב החלומות שלכם
כאן תוכלו להרכיב את המחשב המושלם בדיוק לצרכים שלכם. בין אם אתם גיימרים, יוצרים, מתכנתים או פשוט מחפשים מחשב אמין לשימוש יומיומי – תמצאו אצלנו מבחר רחב של רכיבים שיאפשרו לכם לתכנן מערכת עוצמתית, מותאמת אישית ויעילה.
© 2025 בנו מחשב - כל הזכויות שמורות


בית
אודות
צור קשר
התחבר
הירשם
התחברות
אי-מייל
סיסמה
זכור אותי
הרשמה
כניסה
© 2025 בנו מחשב כל הזכויות שמורות.


בית
אודות
צור קשר
התנתק
מחובר כ: abcd@efg.com
סוג החלק
בחירה
וואט
מחיר
מעבד
65W
₪850
כרטיס מסך
230W
₪2000
לוח אם
55W
₪480
קירור
9W
₪210
ספק כוח
0W
₪320
אחסון
5W
₪260
מארז
0W
₪300
סיכום הרכבה
מחיר כולל: ₪4420
צריכת חשמל: 364W
שמור את ההרכבה
© 2025 בנו מחשב כל הזכויות שמורות.