**תיק פרויקט לפיתוח Web גילי מאייר**

**אפיון ותכנון הפרויקט**

**שם ותיאור הפרויקט:**

**שם הפרויקט:**

Good News **תיאור כללי:** **Good News** הוא אתר אינטרנט שמאפשר למשתמשים לצפות בחדשות חיוביות בלבד.  
המשתמשים יכולים להירשם ולהתחבר למערכת, ולאחר הם יכולים לראות חדשות טובות שקרו.  
האתר כולל ממשק דינמי: כפתורי הניווט משתנים בהתאם למצב ההתחברות (למשל – כפתור "התנתק" מופיע רק למשתמשים מחוברים).  
בנוסף, האתר מספק מידע חיובי, מתוך מטרה לעודד חשיבה חיובית ואופטימית בקרב הגולשים.

**בעיה שהפרויקט פותר:**אנשים רבים מתקשים לשמוע רק חדשות לא טובות בכול יום ולא חדשות טובות בכלל. מערכת זו מספקת להם אפשרות לשמוע רק חדשות טובות כדי לשפר את המצב רוח.

**קהל יעד:**

המערכת מיועדת לאנשים שנמאס להם לשמוע חדשות רעות ושהם רוצים לשמוע ולדעת קצת את החדשות טובות.

**דרישות מערכת:**

**דרישות פונקציונליות:**

דרישות פונקציונליות:

המשתמש יכול להירשם ולהתחבר למערכת

לאחר התחברות, המערכת מציגה רק למשתמש המחובר אפשרות לצפות בחדשות

המשתמש יכול להתנתק מהממשק, ואז הכפתורים "התנתק" נעלמים, ומוצגים שוב "התחבר" ו־"הרשם"

המערכת שומרת את מצב ההתחברות באמצעות session

**דרישות טכניות:**

**Frontend:** HTML, CSS, JavaScript

**Backend:** server.js, Node.js, Express, express-session

**Database:** Node.js, db, db.js, SQLite

**תכנון מבנה האפליקציה**

**מבנה תיקיית הפרויקט**

**מבנה תיקיות:**

/GoodNews  
│── /public                                                                     # HTML, CSS, JS קבצי  
│── /backend                                                                # קבצי צד שרת   
-│ database                                                            # קובץ הגדרות מסד הנתונים   
-│ package.json                                                        # Node.js קובץ תלויות  
│── README.md                                                        # תיעוד הפרויקט

**תרשים זרימה של הניווט:**

תמונה שמכילה טקסט, צילום מסך, תרשים, מלבן

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

**פיתוח צד הלקוח (Frontend)**

**דפי HTML**

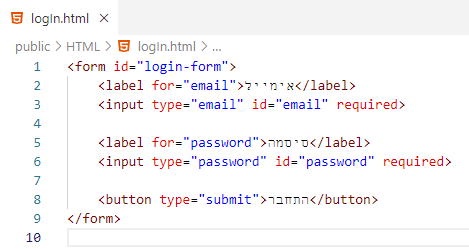
**דפים עיקריים:**

  Index.html – מציג את עמוד הבית

  login.html –מאפשר התחברות

  dashboard.html – מציג את המשימות האישיות

**קטע קוד לדף התחברות (login.html):**



**הסבר מפורט על כל רכיב בקוד:**

 <form>- תגית זו מגדירה טופס HTML לאיסוף נתונים מהמשתמש.

<label> - מספקת תווית לשדה קלט, כך שהמשתמש יבין מה צריך להזין.

 input type="email">> - שדה קלט ייחודי לדואר אלקטרוני.

input type="password">> - שדה קלט לסיסמה, מכסה את הטקסט בנקודות.

required - מבטיח שהמשתמש לא ישאיר את השדה ריק לפני שליחת הטופס.

 button type="submit">> - כפתור לשליחת הנתונים לשרת.

**קוד JavaScript ((Frontend:**קטע קוד script.js (טיפול בהתנהגות כפתורי ניווט לפי התחברות)

- בודק את מצב ההתחברות מהשרת (user-status)

- אם המשתמש מחובר – מציג כפתור 'התנתק' ומסתיר 'התחבר' ו-'הרשם'

- אם לא מחובר – מסתיר את כפתור 'התנתק' ומחזיר את הקישורים להתחברות

קטע מתוך הקוד:

fetch("/user-status")

.then(res => res.json())

.then(data => {

if (data.loggedIn) {

// הסתרת כפתורים והוספת 'התנתק'

} else {

// החזרת כפתורי התחברות

}

});

**פיתוח צד השרת (Backend)**

**יצירת שרת (server.js)**  
 קטעי קוד Node.js ימוקמו תחת סעיף זה בתיק הפרויקט.  
קטע קוד ליצירת שרת server.js:

const session = require("express-session");

app.use(session({

secret: "secretkey",

resave: false,

saveUninitialized: false

}));

* שימוש ב־express-session
* app.get("/user-status", ...) — מחזיר את סטטוס ההתחברות
* app.get("/logout", ...) — מוחק את הסשן ומפנה חזרה לדף הבית
* req.session.user = ... — אחרי התחברות שומר את המשתמש

**תכנון מסד הנתונים**

**מבנה מסד נתונים**  
קטעי קוד מסד הנתונים ימוקמו כאן.  
קוד זה מדגים כיצד המערכת שומרת את מצב ההתחברות של המשתמש ב־session, ומאפשרת להתנתק באמצעות מחיקת הסשן.  
הדבר מאפשר הצגה מותאמת אישית של תוכן למשתמש מחובר בלבד.// התחברות: שמירת המשתמש בסשן

app.post("/login", (req, res) => {

db.get(`SELECT \* FROM users WHERE email = ? AND password = ?`,

[req.body.email2, req.body.password2],

(err, user) => {

if (user) {

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

return res.redirect("/news");

}

res.redirect("/signup");

}

);

});

// התנתקות: מחיקת הסשן

app.get("/logout", (req, res) => {

req.session.destroy(() => res.redirect("/"));

});