**תיק פרויקט לפיתוח Web גילי מאייר**

**אפיון ותכנון הפרויקט**

**שם ותיאור הפרויקט:**

**שם הפרויקט:**

Good News **תיאור כללי:** Good Newsמאפשר למשתמשים לראות חדשות טובות באופן יום יומי, לראות את החדשות הטובות של ימים אחרים, ולקבל עוד ידע מעניין אל ימים מסוימים שלא ידעו לפני.

**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 של אותו יום או יום אחר.

**דרישות מערכת:**

**דרישות פונקציונליות:**

המשתמש יכול להירשם ולהתחבר למערכת

המשתמש יכול לבחור באיזה יום הוא רוצה לראות את החדשות הטובות

המערכת מציגה את החדשות הטובות לפי בחרת היום שהמשתמש בחר

**דרישות טכניות:**

**Frontend:** HTML, CSS, JavaScript

**Backend:** Node.js, Express

**Database:** MongoDB

**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**

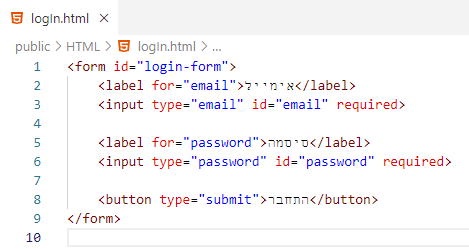
**דפים עיקריים:**

  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:**קטעי קודJavaScript המיועדים להפעלת פונקציונליות בצד הלקוח ימוקמו כאן.  
קטע קוד לטיפול בטופס התחברות script.js)):



**הסבר מפורט:**

 document.getElementById("login-form") - מאתר את טופס ההתחברות לפי מזהה.

 .addEventListener() - מאזין ללחיצה על כפתור שליחה.

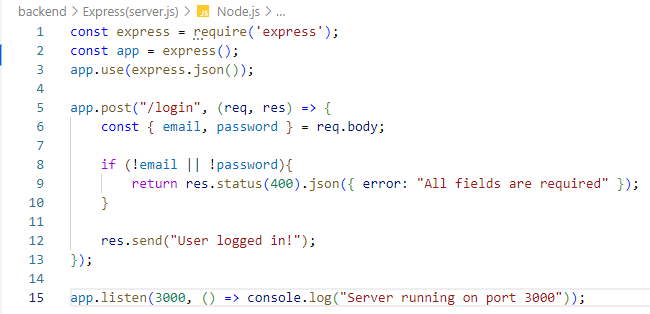
event.preventDefault() - מונע את שליחת הטופס לברירת המחדל.

 בודק אם כל השדות מלאים – אם לא, מציג התראה alert)).

 אם כל הנתונים מלאים – מציג הודעה שההתחברות הצליחה.

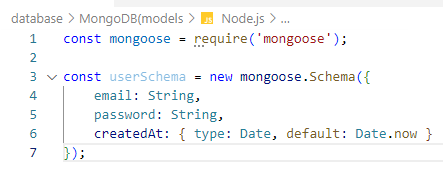
**פיתוח צד השרת (Backend)**

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



**תכנון מסד הנתונים**

**מבנה מסד נתונים**  
קטעי קוד מסד הנתונים ימוקמו כאן.  
קטע קוד למסד נתונים MongoDB (models/User.js):



**בדיקות ואבטחת מידע**

**בדיקות ואבטחת מידע**  
קטעי קוד הקשורים לבדיקה ואבטחה ימוקמו כאן.  
בדיקות טפסים:  
לדוגמה: אם המשתמש השאיר שדה ריק, תוצג הודעת שגיאה.  
אבטחת מידע:  
הצפנת סיסמאות עם :bcrypt

