

## אפיון מערכת

### אתר לתיק עבודות אישי עם אתר פרטי למשתמש

#### CreativePeak – עבודות בגובה אחר

##### תיאור המערכת:

המערכת נועדה להקנות לכל משתמש המתחבר אליה את האפשרות ליצור תיק עבודות אישי, להעלות אליו פרויקטים, וליצור לו מקום אישי באתר שבו יוכל להציג את פרויקטי העבודות שלו. האתר שיתקבל יהיה מורכב מכל תיקי העבודות שהעלו המשתמשים, שם כל אדם יוכל לצפות בכל ולהתרשם מהעבודות שהועלו. המערכת תאפשר ניהול פרויקטים, חיפוש מתקדם, ויכולת לשתף את התיק עבודות בצורה נוחה ומקצועית.

##### מטרת המערכת:

המערכת מיועדת לגרפיקאים/ות המעוניינים להקים תיק עבודות מקצועי וזמין ברשת. כל משתמש יקבל אזור אישי שהוא יכול להעלות אליו פרויקטים ולנהל את תיק העבודות שלו. כמו כן, המערכת תספק כלים ניהוליים עבור המשתמש, המאפשרים לו לשלוט על התכנים שהועלו, לראות סטטיסטיקות ולנהל את הגישה לתוכן שלו.

##### קהל יעד:

- **משתמשים (לקוחות פוטנציאליים):** כל גרפיקאי/ת המעוניין/ת להקים תיק עבודות אישי, ולהפוך אותו לנגיש ללקוחות.
- **לקוחות (גולשים):** כל אדם שיגיע לאתר על מנת לראות את תיקי העבודות של המשתמשים, למצוא פרויקטים שמתאימים להם ולהתעניין בשירותים.

##### פונקציות עיקריות של המערכת:

1. **הגדרת משתמשים ורשום למערכת:**
  - כל משתמש יוכל להירשם ולהתחבר למערכת באמצעות דוא"ל וסיסמה.
  - המערכת תנהל את פרטי המשתמש (שם, אימייל, סיסמה) ותשמור את פרטיהם מאובטחים.
2. **יצירת תיק עבודות אישי:**
  - לאחר ההתחברות, כל משתמש יוכל ליצור תיק עבודות אישי שיכלול:
    - שם תיק העבודות.
    - פרטי משתמש: שם, תיאור קצר על עצמו, ועוד.
3. **העלאת פרויקטים:**
  - כל משתמש יוכל להעלות פרויקטים על ידי:
    - העלאת תמונות.
    - כתיבת תיאור על כל פרויקט.
    - אפשרות לבחירת קטגוריה ותגית לפרויקט.
    - תאריך העלאת הפרויקט.
    - תאריך יצירת הפרויקט
    - ועוד ...

- המערכת תספק ממשק גרפי נוח למנהל הפרויקטים, עם אפשרות להוסיף/לעדכן/למחוק פרויקטים.
- 4. **יצירת אתר אישי עבור כל משתמש:**
  - כל משתמש יקבל מקום באתר מותאם אישית שבו יוצגו כל פרויקטי העבודות שהעלה.
  - יכלול:
    - עיצוב מותאם אישית (המשתמש יוכל לבחור מתוך עיצובים מוגדרים מראש).
    - תצוגה מותאמת של הפרויקטים (כגון גלריה, רשימה, או תצוגה אחרת).
    - פרטי יצירת קשר.
    - כפתור לשיתוף התיק (לדוגמה: שיתוף קישור, כפתור "שלח לי הצעת מחיר").
- 5. **חיפוש ודרכי סינון:**
  - המערכת תאפשר חיפוש מתוחכם של פרויקטים לפי קריטריונים שונים:
    - לפי קהל יעד.
    - לפי תגיות (כגון: "מודרני", "שיווק" וכו').
    - לפי שם פרויקט או שם המשתמש.
    - חיפוש לפי תאריך העלאת פרויקט.
- 6. **סטטיסטיקות ודוחות:**
  - המערכת תספק למשתמשים סטטיסטיקות על צפיות בעבודות שלהם:
    - כמה אנשים צפו בפרויקט.
    - הזמן הממוצע של הצפייה.
- 7. **הגדרות הרשאות וגישה:**
  - כל משתמש יוכל להגדיר את ההרשאות לתיק העבודות שלו:
    - גישה ציבורית: פרויקטים גלויים לכולם.
    - גישה פרטית: רק למי שיש לו קישור אישי או הרשאות גישה.
    - גישה למזמינים פוטנציאליים: אישור לצפייה רק עבור משתמשים שנרשמו או שילמו.
- 8. **שיתוף ותיאום עם לקוחות פוטנציאליים:**
  - כל משתמש יוכל לשתף את תיק העבודות עם לקוחות פוטנציאליים באמצעות קישור אישי.
  - המערכת תכלול אפשרויות ליצירת קשר (כגון טופס יצירת קשר, כפתור WhatsApp או דואר אלקטרוני).

## דרישות טכנולוגיות:

- **Backend:** ע"י Node.js + Express, לפיתוח API לניהול פרויקטים, משתמשים וסטטיסטיקות.
- **Frontend:**
  - ע"י React לפיתוח ממשק משתמש אינטראקטיבי - למשתמשים, עם שימוש ב-Redux לניהול מצב.
  - ע"י Angular לפיתוח ממשק משתמש ללקוחות הצופים בתיקי העבודות.
- **Database:** ע"י MySQL לאחסון נתוני משתמשים, פרויקטים, סטטיסטיקות.
- **אחסון קבצים:** שימוש ב-Amazon S3 לאחסון תמונות ונתוני פרויקטים.
- **Authentication:** ע"י JWT (JSON Web Token) לאימות והרשאות גישה.
- **Responsive Design:** עיצוב רספונסיבי שיתמוך בצפייה במובייל ובדסקטופ.

## ממשק משתמש: (UI/UX)

### 1. דף התחברות והרשמה:

- טופס רישום והתחברות לאתר עבור משתמשים חדשים/קיימים.
- אפשרות לשחזור סיסמה במקרה של שכחה.

### 2. דף אישי למשתמש (Dashboard):

- הצגת תיק העבודות של המשתמש.
- כפתור להוספת פרויקט חדש.
- אפשרות לערוך או למחוק פרויקטים קיימים.
- ניהול סטטיסטיקות צפיות לכל פרויקט.

### 3. דף פרויקט אישי:

- הצגת פרטי הפרויקט כולל תמונות, תיאור, תגיות, קטגוריות.
- כפתור "שיתוף" לשליחת הקישור ללקוחות.
- מידע על הצפיות בפרויקט.

### 4. דף תיק העבודות של המשתמש - הלקוח:

- הצגת תיק העבודות בצורה מקצועית ואטרקטיבית.
- אפשרות לסינון פרויקטים לפי קטגוריות, תגיות ודרכים אחרות.

## אפיון תמציתי

### ■ רשימת פונקציות המערכת

#### 1. ניהול משתמשים:

- Route: POST /api/users/register
- Route: POST /api/users/login
- Route: POST /api/users/profile

#### 2. ניהול פרויקטים:

- Route: POST /api/projects/add
- Route: GET /api/projects/{id}
- Route: DELETE /api/projects/{id}
- פרמטרים:

Body: { title, description, category, url\_project, created\_at }

○ פלט: { projectId, url }

○ לוגיקה: אימות קובץ תמונה, גודל מקסימלי מוגבל.

#### 3. חיפוש וסינון פרויקטים:

- Route: GET /api/projects/search?query=...
- פרמטרים: { category, year, client }
- פלט: { projects: [...] }

לוגיקה: חיפוש ומיון לפי פרמטרים שהוזנו.

הגדרת מסד נתונים (Database Schema) :

1. הטבלאות הדרושות

- Users (משתמשים):** אנשים שיש להם תיק עבודות.
- Files (קבצים):** קבצים שהעלו המשתמשים (כמו תמונות, קבצי גרפיקה, וכו').
- Categories (קטגוריות):** סוגי עבודות שיכולים להיות מקוטלגים (למשל: לוגואים, עיצוב אתרים, פרסומות).
- Tags (תגים):** תיוגים שניתן להחיל על קבצים (למשל: "מודרני", "נוסטלגי").

2. טבלאות ועמודות

טבלת Users (משתמשים):

שם העמודה	סוג נתונים	מאפיינים	תיאור
id	INT	PK	מזהה ייחודי למשתמש
first_name	VARCHAR(50)	NOT NULL	שם פרטי
last_name	VARCHAR(50)	NOT NULL	שם משפחה
email	VARCHAR(100)	UNIQUE	אימייל
password	VARCHAR(255)	NOT NULL	סיסמה
created_at	DATETIME	DEFAULT NOW()	תאריך יצירה
updated_at	DATETIME	DEFAULT NOW()	תאריך עדכון אחרון
updated_by	INT	FK (Users)	מזהה המשתמש שעדכן את הנתונים

טבלת Files (קבצים):

שם העמודה	סוג נתונים	מאפיינים	תיאור
id	INT	PK	מזהה ייחודי לקובץ
user_id	INT	FK (Users)	מזהה המשתמש שיצר את הקובץ
file_name	VARCHAR(255)	NOT NULL	שם הקובץ
description	VARCHAR(255)	NOT NULL	תיאור
file_path	VARCHAR(255)	NOT NULL	מיקום הקובץ במערכת
category_id	INT	FK (Categories)	מזהה הקטגוריה של הקובץ
created_at	DATETIME	DEFAULT NOW()	תאריך יצירה
created_by	INT	FK (Users)	מזהה המשתמש שיצר את הקובץ
updated_at	DATETIME	DEFAULT NOW()	תאריך עדכון אחרון

updated_by	INT	FK (Users)	מזהה המשתמש שעדכן את הנתונים
------------	-----	------------	------------------------------

## טבלת Categories (קטגוריות):

שם העמודה	סוג נתונים	מאפיינים	תיאור
id	INT	PK	מזהה ייחודי לקטגוריה
name	VARCHAR(100)	NOT NULL	שם הקטגוריה
description	TEXT	NULL	תיאור הקטגוריה

## טבלת Tags (תגיות):

שם העמודה	סוג נתונים	מאפיינים	תיאור
id	INT	PK	מזהה ייחודי לתג
name	VARCHAR(100)	NOT NULL	שם התג

## טבלת FileTags (טבלת חיבור בין קבצים לתגים):

שם העמודה	סוג נתונים	מאפיינים	תיאור
file_id	INT	FK (Files)	מזהה הקובץ
tag_id	INT	FK (Tags)	מזהה התג

## טבלת Views (סטטיסטיקות צפייה):

שם העמודה	סוג נתונים	מאפיינים	תיאור
id	INT	PK	מזהה ייחודי לצפייה
file_id	INT	FK (Tags)	מזהה הקובץ
user_id	INT	FK (Tags)	מזהה המשתמש הצופה
date	DATETIME	DEFAULT NOW()	תאריך ושעת צפיה

## 4. קשרים בין הטבלאות

- **One-to-Many:**
  - **Users ↔ Files** – משתמש אחד יכול להעלות קבצים רבים.
  - **Categories ↔ Files** – קטגוריה אחת יכולה להכיל מספר קבצים.
- **Many-to-Many:**
  - **Files ↔ Tags** – קובץ אחד יכול להיות משויך למספר תגים, ותג אחד יכול להיות משויך למספר קבצים. נדרשת טבלת חיבור **FileTags**.
- **Many-to-One:**
  - **Files ↔ Views** – הרבה צפיות יכולות להיות לאותו הקובץ.

- **Users ↔ Views** – הרבה צפיות יכולות להיות לאותו לקוח.

## 5. תרשים קשרים (ERD)

ליצירת תרשים ERD, זה יכול להיראות כך:

1. **Users ↔ Files** – many-קשר אחד-ל (User uploads many files).
2. **Files ↔ Categories** – many-קשר אחד-ל (File belongs to one category).
3. **Files ↔ Tags** – many-קשר many-ל-many (Files can have many tags, tags can be assigned to many files).
4. **Files ↔ Views** – many-קשר אחד-ל (Views uploads many files).
5. **Users ↔ Views** – many-קשר אחד-ל (Views uploads many users).

## תרשים מבנה המערכת (Architecture Diagram) :

המערכת תתבסס על אדריכלות MVC (Model-View-Controller) ותחולק לשלושה חלקים עיקריים:

1. **Backend (צד שרת):**
  - Node.js + Express לניהול ה-API.
  - JWT לניהול הרשאות ומאגרי נתונים.
  - MySQL לאחסון הנתונים.
2. **Frontend (צד לקוח):**
  - React לפיתוח ממשק משתמש אינטראקטיבי.
  - שימוש ב-Redux לניהול מצב המערכת.
  - עיצוב רספונסיבי שיתמוך במכשירים שונים.
3. **אחסון קבצים:**
  - Amazon S3 לאחסון קבצים - תמונות ושיתוף הקבצים בין המערכת למשתמשים.
4. **Security:**
  - JWT לאימות והגנה על המידע.
  - הצפנת סיסמאות בעזרת ספריות כמו bcrypt.

## תיאור של כל תהליך במערכת:

1. **תהליך ההרשמה וההתחברות:**
  - משתמש חדש יוכל להירשם על ידי הזנת שם, אימייל, וסיסמה.
  - אחרי ההרשמה, הוא יוכל להתחבר למערכת עם פרטי ההתחברות שסיפק.
  - משתמשים יכולים לשחזר את הסיסמה אם הם שוכחים אותה.
2. **יצירת תיק עבודות אישי:**
  - לאחר ההתחברות, המערכת תספק למשתמש פאנל ניהול אישי.
  - המשתמש יוכל להוסיף פרויקטים חדשים לתיק העבודות, עם תמונות ותיאורים.
  - המערכת תאפשר למשתמש לשייך פרויקטים לקטגוריות ולתגיות כדי להקל על הגישה אליהם.

**3. יצירת מקום אישי באתר תיקי העבודות:**

- כל משתמש יקבל אתר אישי מותאם אישית, שיכלול את תיק העבודות שלו.
- האתר יציג את הפרויקטים שהעלו המשתמשים, ויתעדכן אוטומטית בכל פעם שהמשתמש מוסיף פרויקט חדש.
- האתר יוכל לכלול עיצוב מותאם אישית, והמשתמש יוכל לבחור בין מספר תבניות עיצוב מוכנות מראש.

**4. חיפוש וסינון תיקי עבודות ופרויקטים:**

- המערכת תספק ממשק חיפוש מתקדם שיאפשר למשתמשים לחפש תיקי עבודות לפי קריטריונים כגון: קטגוריות, תגיות, או תאריך.
- החיפוש יהיה ממוקד ומאפשר סינון מדויק של תיקי העבודות והפרויקטים.

**5. סטטיסטיקות ודוחות:**

- המערכת תספק למשתמשים סטטיסטיקות על פרויקטי העבודות שלהם, כמו מספר הצפיות.
- כל משתמש יוכל לראות דוחות ולהבין אילו פרויקטים הכי פופולריים ואילו מבין העבודות שלו מעוררות יותר עניין.

**6. הגדרות פרטיות וגישה:**

- המערכת תאפשר למשתמשים להגדיר את פרטיות הפרויקטים שלהם:
  - גישה חופשית (לכולם).
  - גישה מוגבלת (רק למי שיש לו קישור ישיר או הרשאה).
  - הצגת פרויקטים רק עבור לקוחות שנרשמו או שילמו.

**7. שיתוף עם לקוחות פוטנציאליים:**

- כל משתמש יכול לשתף את תיק העבודות עם אחרים על ידי שליחת קישור ייחודי או בחירה בכפתור שיתוף.
- בנוסף, המערכת תאפשר הוספת כפתורי יצירת קשר ישירים כמו דואר אלקטרוני, טלפון, או טופס יצירת קשר.

**סבבי פיתוח (ספרינטים):****ספרינט 1: תכנון והקמה ראשונית**

- יצירת בסיס נתונים (Database) בעזרת MySQL .
- יצירת פרויקט למשתמשים בשפת React.
- יצירת פרויקט ללקוחות בשפת Angular.
- העלאת הפרויקטים לגיט.
- פתיחת חשבון במערכת Amazon להעלאת הפרויקט לענן.

**ספרינט 2: פיתוח פאנל ניהול למשתמש**

- הקמת ממשק התחברות ורישום משתמשים: פיתוח API לרישום והתחברות משתמשים.

- **הקמת מערכת פרויקטים API**: לניהול פרויקטים, כולל הוספה, עריכה ומחיקה.
- **הקמת מערכת הרשאות בסיסית**: הרשאות צפייה והעלאת פרויקטים.
- **ממשק ניהול אישי**: עיצוב ויישום פאנל משתמש, כולל אפשרות להוסיף פרויקטים, לצפות בסטטיסטיקות ולנהל את תיק העבודות.
- **העלאת פרויקטים**: תמיכה בהעלאת קבצים (תמונות) וכתובת תיאורים.

### ספרינט 3: פיתוח מנגנוני חיפוש וסינון

- **חיפוש פרויקטים**: יצירת מנגנון חיפוש מתקדם (לפי קטגוריות, תגיות, שם פרויקט, תאריך וכו').
- **מערכת סטטיסטיקות**: הפקת דוחות ומעקב אחרי צפיות וסטטיסטיקות עבור כל פרויקט.
- **שיפור UI**: עיצוב רספונסיבי לשימוש על דסקטופ.

### ספרינט 4: אבטחה ושיפור ביצועים

- **אבטחת המידע**: הצפנת סיסמאות והגנה על המידע הרגיש באמצעות JWT.
- **שיפור ביצועים**: אופטימיזציה של טעינת דפים והעלאות.
- **פריסת המערכת**: פריסת המערכת בסביבת ענן.
- **בדיקות ופריסת המערכת**: בדיקות פונקציונליות ואוטומטיות של המערכת (Unit Testing, Integration Testing).

### סיכום:

המערכת שהוגדרה כאן נועדה להקנות למשתמשים את הכלים להקים תיק עבודות אישי ברשת בצורה קלה, מקצועית ויעילה. המערכת תספק לכל משתמש אתר אישי המאפשר להציג את פרויקטי העבודות שלו, וללקוחות פוטנציאליים תתאפשר גישה נוחה ומסודרת לצפייה בפרויקטים. תהליך העלאת התכנים, הצפייה והסטטיסטיקות הפנימיות במערכת יאפשר למשתמשים להתנהל בצורה מקצועית וגמישה עם האפשרות לניהול פרטיות וגישה.