

פרויקט סיום בקורס 20503

סדנה בתכנות מתקדם בשפת Java

שם הפרויקט:

SynchBoard



מגיש: שגיא מנחם, 208645937.

תאריך הגשה: 23/09/2025.

1. מבוא:

SynchBoard הוא יישום Full-Stack המשמש כלוח ציור שיתופי, המיועד לשיתוף פעולה ועבודה ויזואלית בצוותים בזמן אמת. היישום מאפשר למספר משתמשים לצייר, לשרטט ולתקשר על גבי לוחות דיגיטליים משותפים, עם סנכרון מיידי של כל פעולה. הפלטפורמה מיועדת לצוותים, אנשי חינוך וארגונים הזקוקים לגלי עבודה ויזואלי ואינטראקטיבי. הפרויקט כולל מערכת מקיפה של אימות משתמשים, ניהול לוחות, צ'אט מובנה, כלי ציור וניהול הרשאות לחברי הלוח. המערכת פותחה בטכנולוגיות Web מודרניות ומציעה גישה מבוססת דפדפן ואפשרות הרצה מקומית באמצעות קונטיינרים של Docker.

תכונות מרכזיות:

- ציור שיתופי בזמן אמת: עריכה סימולטנית של מספר משתמשים עם סנכרון מיידי.
- מערכת אימות משתמשים מקיפה: כולל אימות באמצעות אימייל, טיפוס סיסמה והתחברות דרך Google (OAuth2).
- ניהול לוחות מתקדם: יצירה, הזמנת חברים ובקרת גישה מבוססת הרשאות.
- כלי ציור עשירים: מברשות, צורות, קווים, טקסט, בחירת צבעים, הגדרות עובי קו ופונקציונליות Undo / Redo.
- צ'אט מובנה: מערכת צאט בזמן אמת המשולבת בכל לוח ומאפשרת תקשורת צוותית.
- עיצוב רספונסיבי ותמיכה בריבוי שפות: תמיכה מלאה במחשבים שולחניים ובטאבלטים, עם ממשק מלא באנגלית ובעברית (כולל תמיכה בRTL).
- ניהול קבצים: יכולות העלאת תמונות ללוח, תמונות פרופיל וייצוא של לוחות שהושלמו.

2. מדריך למשתמש:

תפקידי משתמשים:

למערכת SynchBoard שלושה סוגי משתמשים, שלכל אחד מהם הרשאות שונות:

• אורח:

- יכול לגשת רק לעמוד ההתחברות וההרשמה.
- אינו יכול לצפות בלוחות, להצטרף אליהם או להשתתף בסביבת העבודה השיתופית.
- חייב להשלים תהליך הרשמה (ואימות אימייל) כדי לקבל גישה לאפליקציה.

• משתמש רשום:

- יכול ליצור לוחות חדשים ולהשתתף באופן מלא בסביבת העבודה השיתופית.
- בעל גישה לכל כלי הציור, שליחת הודעות בצ'אט, עריכת פרופיל אישי וניהול העדפות אישיות (ערכת נושא, כלים וכו').
- יכול לעזוב לוחות שבהם אינו מנהל.

• מנהל לוח:

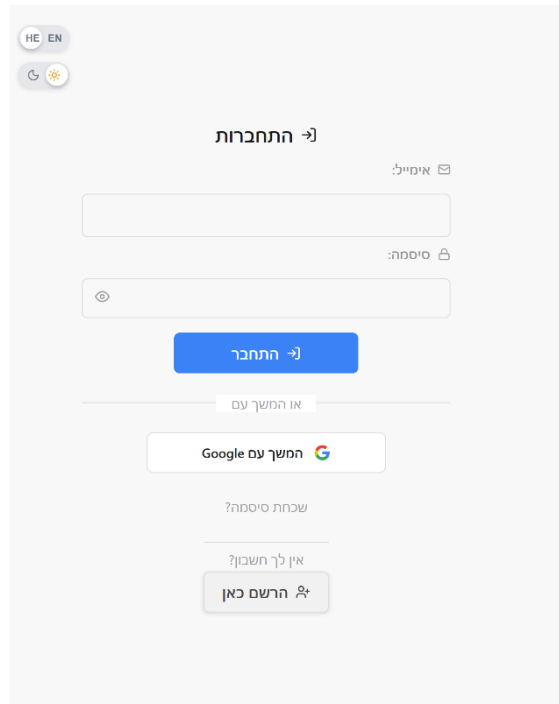
- בעל הרשאות של משתמש רשום.
- בנוסף, יכול להזמין חברים חדשים ללוח, להסיר חברים קיימים, לקדם משתמשים אחרים למעמד מנהל, ולשנות את הגדרות הלוח (שם, תיאור, מידות קנבס וצבע רקע).

מדריך למשתמש

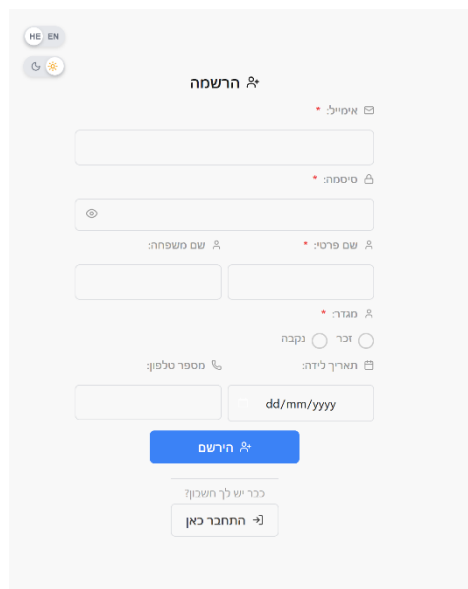
תהליכי עבודה מרכזיים:

- הרשמת משתמש חדש:

(1) מעבר לטופס ההרשמה: בעמוד האימות, לחץ על לשונית Register או על כפתור Create Account כדי לעבור מטופס ההתחברות לטופס ההרשמה.

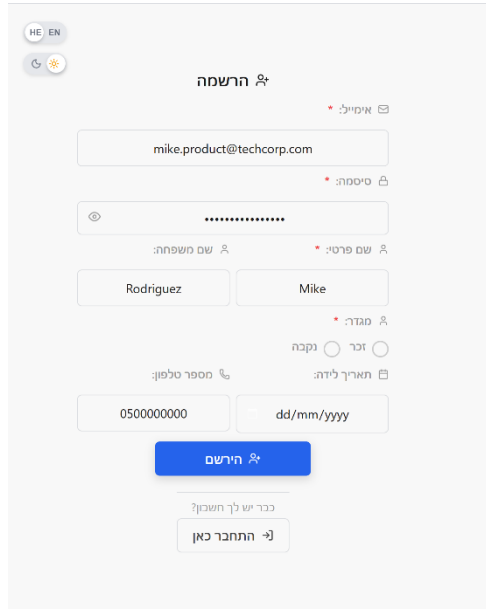


(2) מילוי פרטים: מלא את שדות החובה בטופס: כתובת אימייל, סיסמה, שם פרטי ומגדר. ניתן להוסיף גם שם משפחה, מספר פלאפון ותאריך לידה.



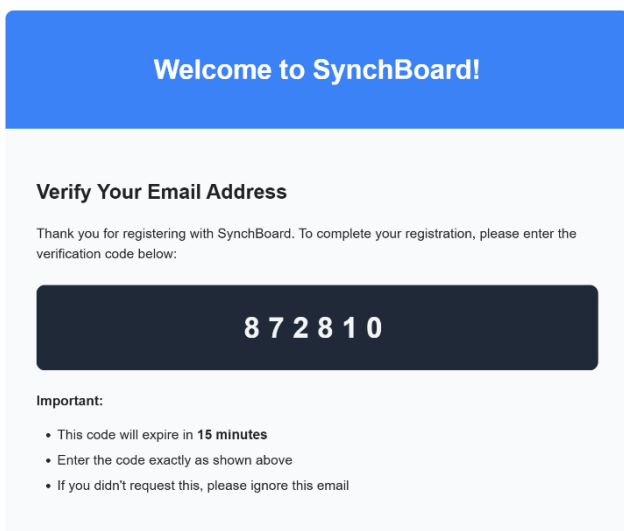
מדריך למשתמש

(3) שליחת הטופס: לחץ על הכפתור 'הירשם' כדי לשלוח את טופס ההרשמה ולהתחיל את תהליך יצירת החשבון.



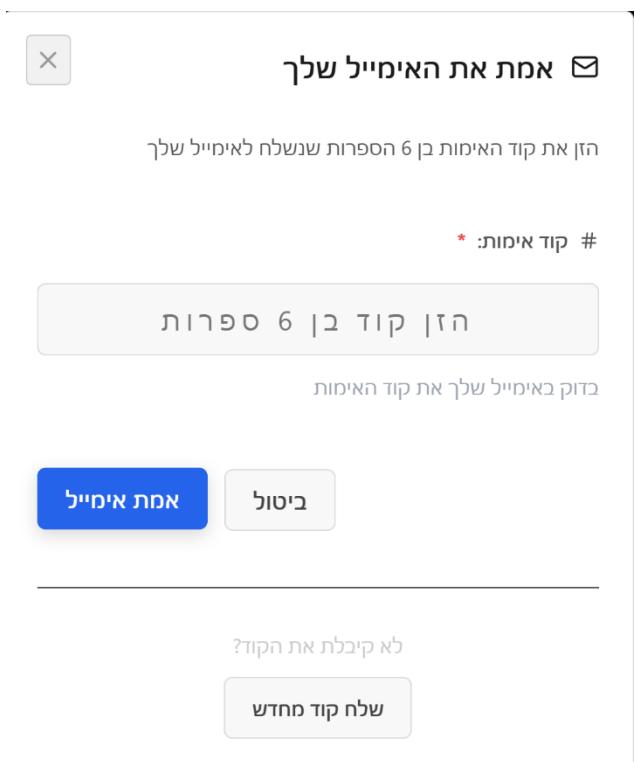
The registration form is titled 'הירשמה' (Registration). It includes fields for email (mike.product@techcorp.com), password (masked with dots), first name (Rodriguez), and last name (Mike). There are also fields for phone number (0500000000) and date of birth (dd/mm/yyyy). A blue button labeled 'הירשם' (Register) is at the bottom. Below the button, there is a link 'כבר יש לך חשבון?' (Already have an account?) and a link 'לחצו כאן' (Click here).

(4) אימות אימייל: בדוק את תיבת הדואר הנכנס שלך וחפש קוד אימות בן 6 ספרות (ייתכן שיגיע לספאם). הזן את הקוד בחלון האימות שיופיע באופן אוטומטי.



The email verification screen has a blue header with the text 'Welcome to SynchBoard!'. Below it, the title 'Verify Your Email Address' is followed by a message: 'Thank you for registering with SynchBoard. To complete your registration, please enter the verification code below:'. A large black box displays the code '872810'. Below this, an 'Important:' section lists three points: 'This code will expire in 15 minutes', 'Enter the code exactly as shown above', and 'If you didn't request this, please ignore this email'.

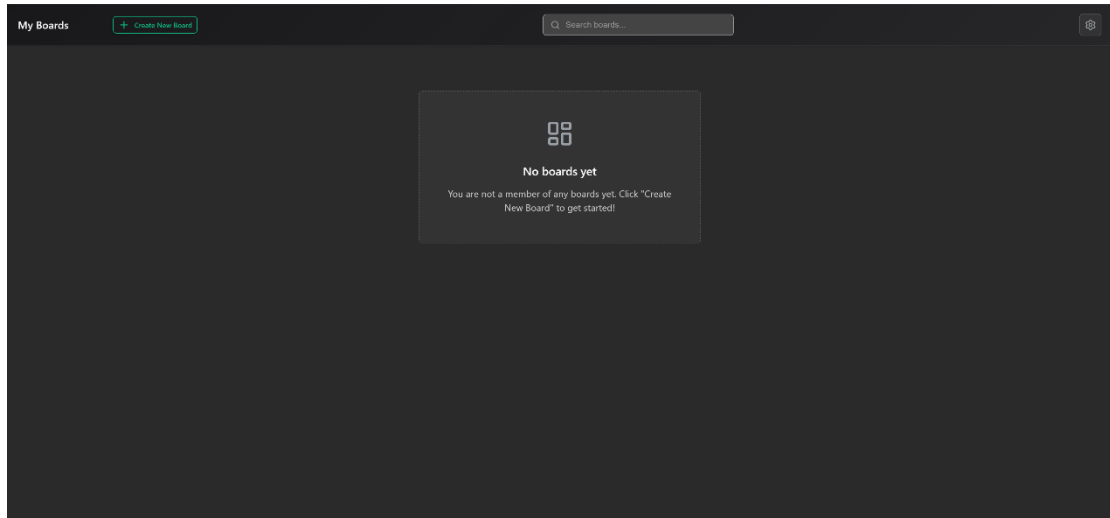
This is an automated message, please do not reply to this email.



The email verification modal has a title bar with a close button and the text 'אמת את האימייל שלך' (Verify your email). The main content area says 'הזן את קוד האימות בן 6 הספרות שנשלח לאימייל שלך' (Enter the 6-digit verification code sent to your email). Below this, there is a label '# קוד אימות: *' and a large input field with the placeholder 'הזן קוד בן 6 ספרות'. A blue button labeled 'אמת אימייל' (Verify email) and a grey button labeled 'ביטול' (Cancel) are at the bottom. Below the buttons, there is a link 'לא קיבלת את הקוד?' (Didn't receive the code?) and a button 'שלח קוד מחדש' (Resend code).

מדריך למשתמש

(5) סיום והכוונה: לאחר אימות מוצלח, תועבר אוטומטית לעמוד רשימת הלוחות, שם תוכל להתחיל ליצור לוחות חדשים.

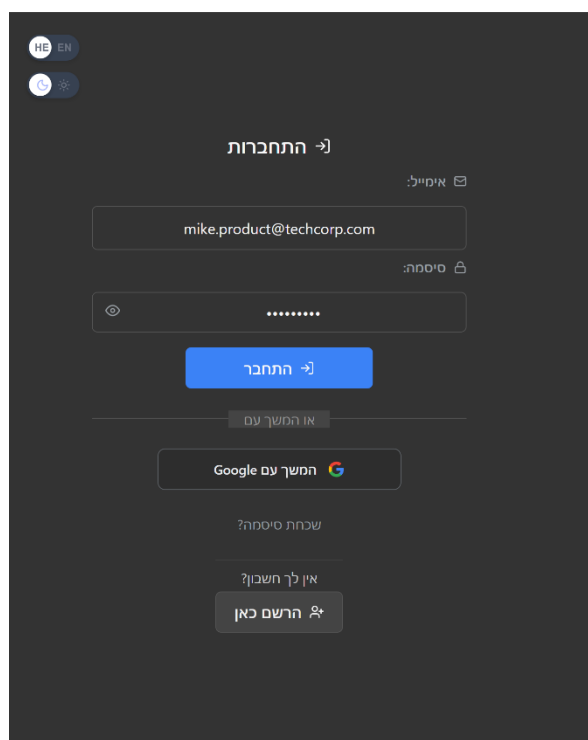


• התחברות למערכת

תהליך כניסה למערכת עם חשבון קיים.

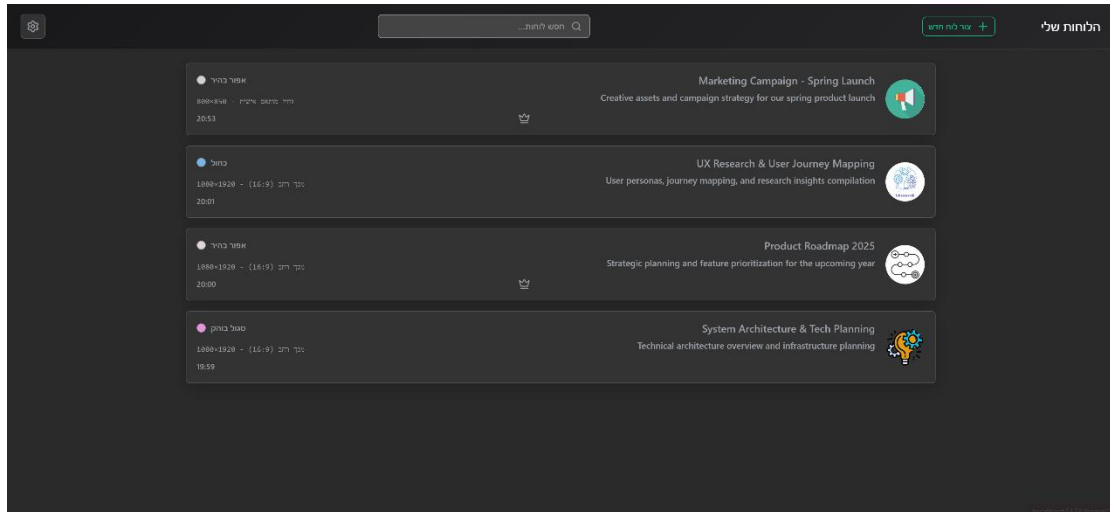
(1) הזנת אימייל: בעמוד האימות, ודא שלשונית 'Login' פעילה והזן את כתובת האימייל הרשומה שלך.

(2) הזנת סיסמה: הזן את הסיסמה בשדה המיועד. ניתן להשתמש באייקון העין כדי להציג או להסתיר את הסיסמה.



מדריך למשתמש

- (3) אימות: לחץ על כפתור 'התחבר' כדי להתחבר. לחלופין, אם נרשמת עם Google לחץ על כפתור 'המשך עם Google'.
- (4) הכוונה לעמוד הראשי: עם התחברות מוצלחת, תועבר אוטומטית לעמוד רשימת הלוחות שלך.



- (5) שכחת סיסמה: אם שכחת את הסיסמה, לחץ על הקישור 'שכחת סיסמה?' מתחת לטופס ההתחברות כדי להתחיל תהליך איפוס סיסמה באמצעות המייל.

×

שכחת סיסמה

הזן את האימייל שלך כדי לקבל קוד איפוס סיסמה

אימייל: *

הזן את כתובת האימייל שלך

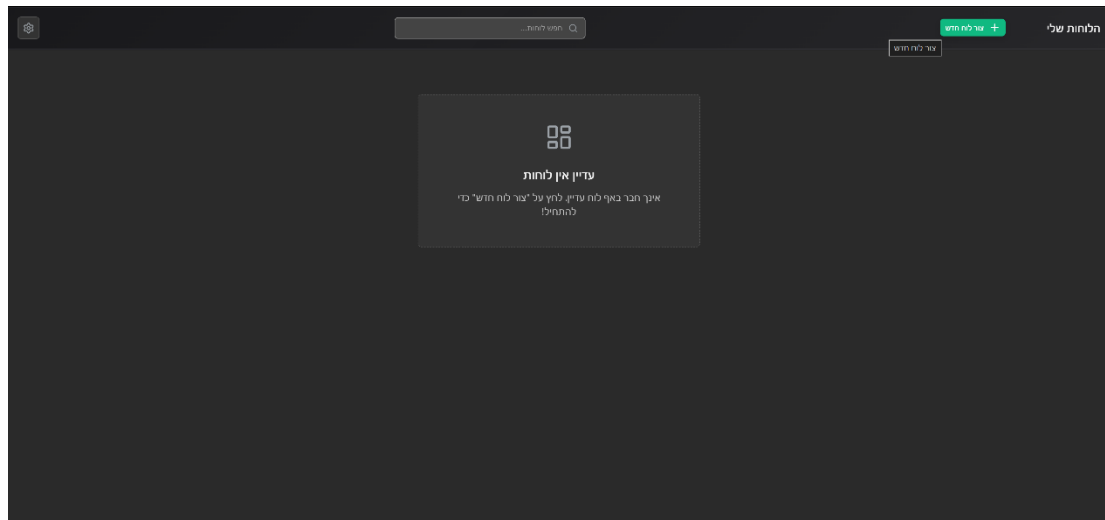
שלח קוד איפוס

ביטול

מדריך למשתמש

• יצירת לוח חדש

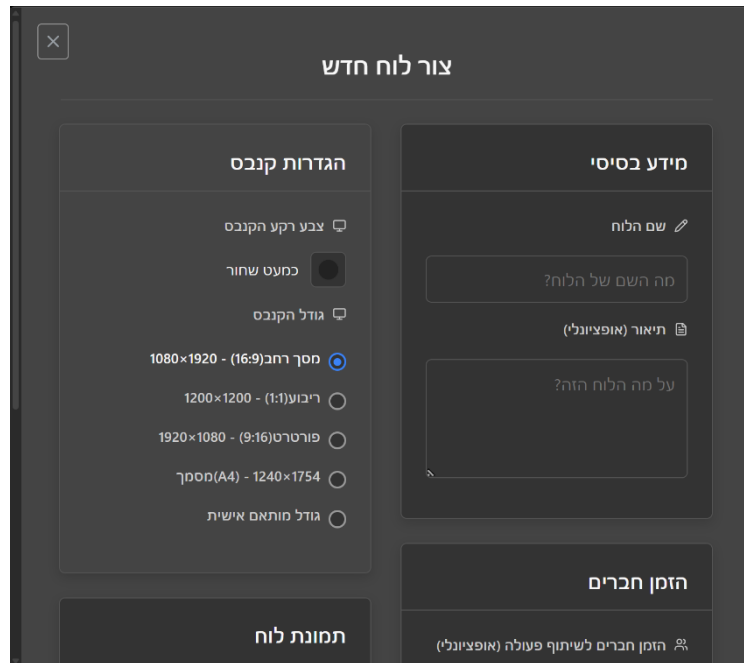
- 1) לחיצה על יצירת לוח: מעמוד רשימת הלוחות, לחץ על הכפתור '+'
צור לוח חדש' בסרגל הכלים.



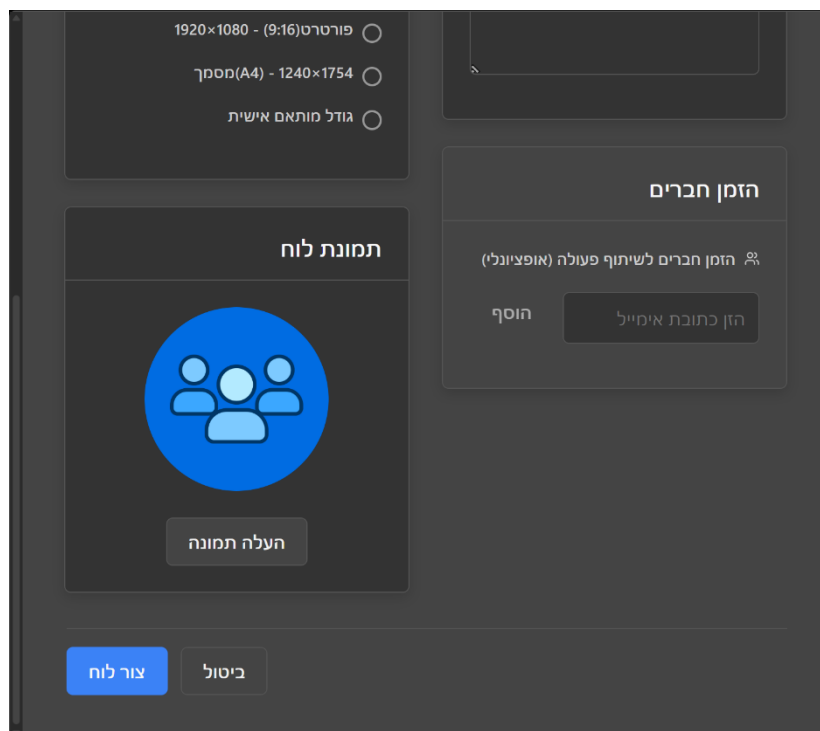
- 2) הגדרת שם ותיאור: בחלון שנפתח, הזן שם ללוח (100-3 תווים, חובה) ותיאור אופציונלי.

מדריך למשתמש

3) הגדרות הקנבס: קבע את הגדרות משטח העבודה: רוחב (-400 4000 פיקסלים), גובה (400-4000 פיקסלים) וצבע רקע באמצעות בורר הצבעים.



4) הגדרות נוספות (אופציונלי): ניתן להעלות תמונת נושא ללוח (עד 10MB) ולהזמין חברים על ידי הזנת כתובות האימייל שלהם.

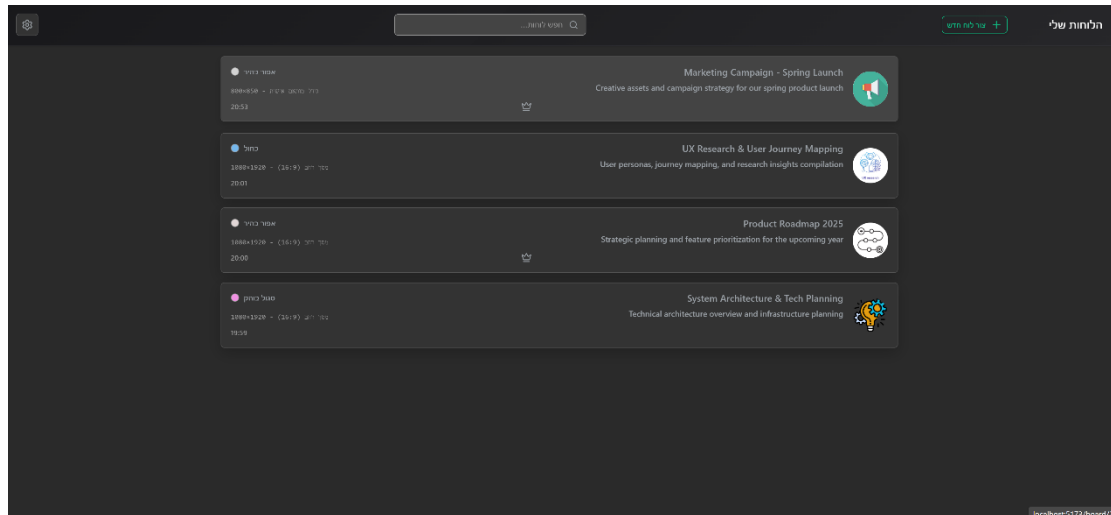


מדריך למשתמש

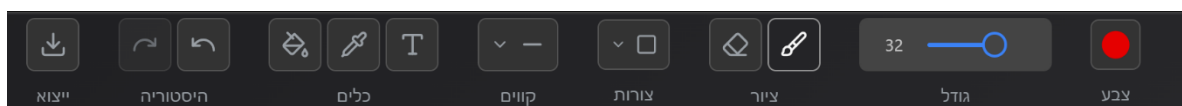
(5) סיום יצירה: לחץ על הכפתור 'צור לוח' כדי לסיים את התהליך.
תועבר אוטומטית לסביבת העבודה של הלוח החדש.

• עבודה על לוח שיתופי:

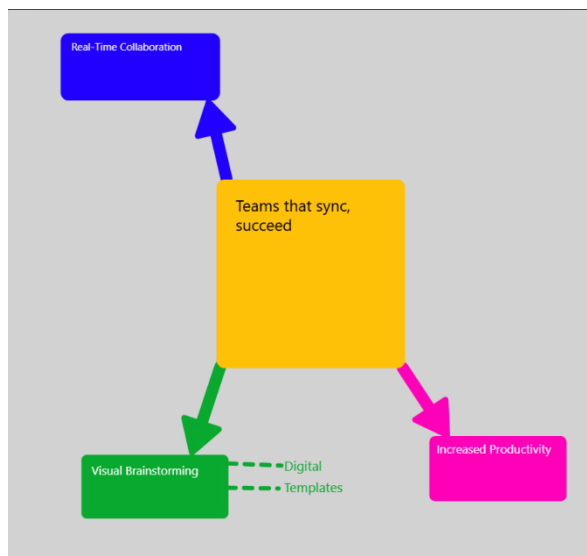
(1) כניסה ללוח: לחץ על כרטיסיית הלוח הרצוי מרשימת הלוחות שלך.



(2) בחירת כלי ציור: בסביבת העבודה של הלוח, בחר כלי ציור מסרגל הכלים: מברשת, צורות, קווים, כלי טקסט או מחק. התאם את צבע ועובי הקו.

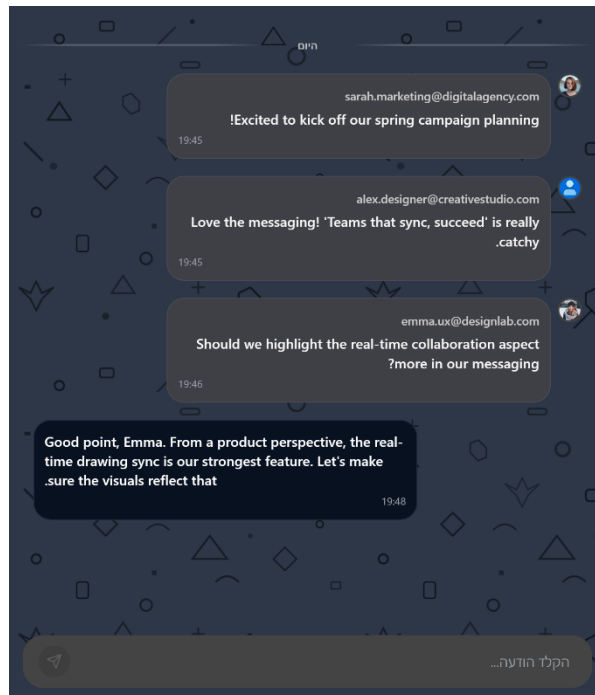


(3) ציור על הקנבס: התחל לצייר על הקנבס באמצעות לחיצה וגרירה של העכבר.

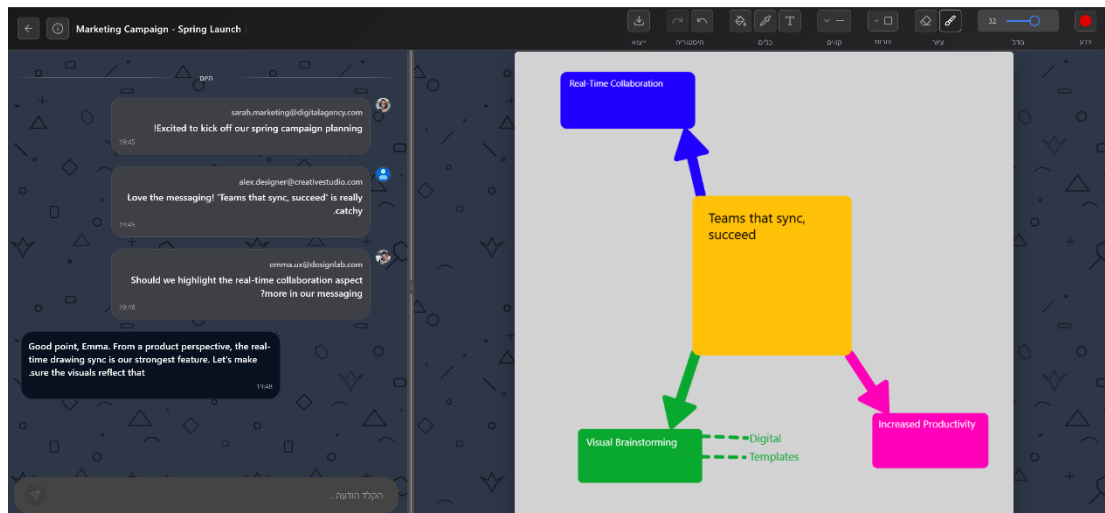


מדריך למשתמש

(4) שימוש בצ'אט: השתמש בחלונית הצ'אט בצד ימין כדי לתקשר עם חברי הצוות בזמן אמת.



(5) פעולות נוספות: השתמש בכפתורי Undo / Redo בסרגל הכלים כדי לבטל או לשחזר פעולות. השתמש בכפתור ההורדה כדי לייצא את מצב הלוח הנוכחי לקובץ תמונה.



3. תיעוד טכני:

מטרה	גרסה	טכנולוגיה	תחום
ספריית UI ראשית לבניית ממשק משתמש אינטראקטיבי.	19.1.1	React	צד לקוח
פיתוח עם בטיחות טיפוסים לשיפור חווית הפיתוח.	5.9.2	TypeScript	
כלי בנייה ושרת פיתוח מודרני.	7.0.0	Vite	
ניהול ניתוב צד לקוח באפליקציית Single-Page.	7.6.3	React Router	
פרה פרוססור ל CSS עם תמיכה במודולים לעיצוב קומפוננטות.	1.91.0	SCSS	
תקשורת WebSocket באמצעות פרוטוקול STOMP.	7.1.1	@stomp/stompjs	
קליינט HTTP לתקשורת עם RESTful API.	1.10.0	Axios	
ספריה לניהול תרגומים (אנגלית / עברית) ותמיכה ב-RTL.	15.6.0	React i18next	
שפת התכנות הראשית לפיתוח צד שרת.	24	Java	צד שרת
Framework ראשי ליישום, מספר הזרקת תלויות ותכונות Production-ready.	3.5.5	Spring Boot	
ניהול אבטחה, אימות, הרשאות, טוקני JWT ואינטגרציית OAuth2.	כלול	Spring Security	
תמיכה ב WebSocket עם STOMP לתקשורת בזמן אמת.	כלול	Spring WebSocket	
שכבת גישה לנתונים עם תבנית Repository ושאלות אוטומטיות.	כלול	Spring Data JPA	
מיפוי אובייקטי רלציוני לניהול ישויות בבסיס הנתונים.	כלול	Hibernate	
אינטגרציה עם שירות שליחת אימיילים לאימות ואיפוס סיסמה.	4.10.3	SendGrid	
יצירה ואימות של טוקני אימות Stateless.	0.13.0	JSON Web Tokens	
כלי אוטומציה לבנייה וניהול תלויות.	כלול	Gradle	
בסיס נתונים רלציוני ראשי לאחסון נתוני משתמשים, לוחות, אובייקטים ועוד.	17	PostgreSQL	בסיס נתונים
פלטפורמת קונטיינריזציה להרצה עקבית בסביבות שונות.	Latest	Docker	Deployment
תזמור של מספר קונטיינרים.	Latest	Docker Compose	
שרת Web להגשת קבצים סטטיים של Frontend.	Latest	Nginx	

Message Broker לתקשורת WebSocket והפצת אירועים בזמן אמת.	2.31.0	ActiveMQ Artemis	
--	--------	------------------	--

טכנולוגיות בשימוש:

ארכיטקטורה:

- תבנית עיצוב: הפרויקט נבנה כמערכת אחת (מונולית), אך תוכנן מראש כך שיהיה קל לפרק חלקים ממנו בעתיד למיקרו שירותים. המערכת מחלוקת לשלושה חלקים עיקריים כדי להקל על הפיתוח – מודל, תצוגה ובקר.
- תקשורת:
 - RESTful API: עבור בקשות HTTP סטנדרטיות, עם העברת נתונים בפורמט JSON.
 - WebSocket (STOMP): עבור תקשורת דו כיוונית בזמן אמת, המשמשת לסנכרון פעולות הציור והודעות הצ'אט.
- מבנה הפרויקט:
 - Frontend: הקוד מאורגן לפי פיצ'רים, כאשר כל פיצ'ר מכיל את הקומפוננטות, ההוקים, השירותים והעמודים הרלוונטיים לו. תיקיית shared מכילה קוד לשימוש חוזר כמו קומפוננטות UI, כלי עזר וקבועים.
 - Backend: הפרויקט עם הפרדת שכבות ברורה: entity (ישויות JPA), repository (שכבת גישה לנתונים), service (לוגיקה עסקית), controller (נקודות קצה של ה-API), config (הגדרות תצורה), ו-dto (אובייקטי העברת נתונים).

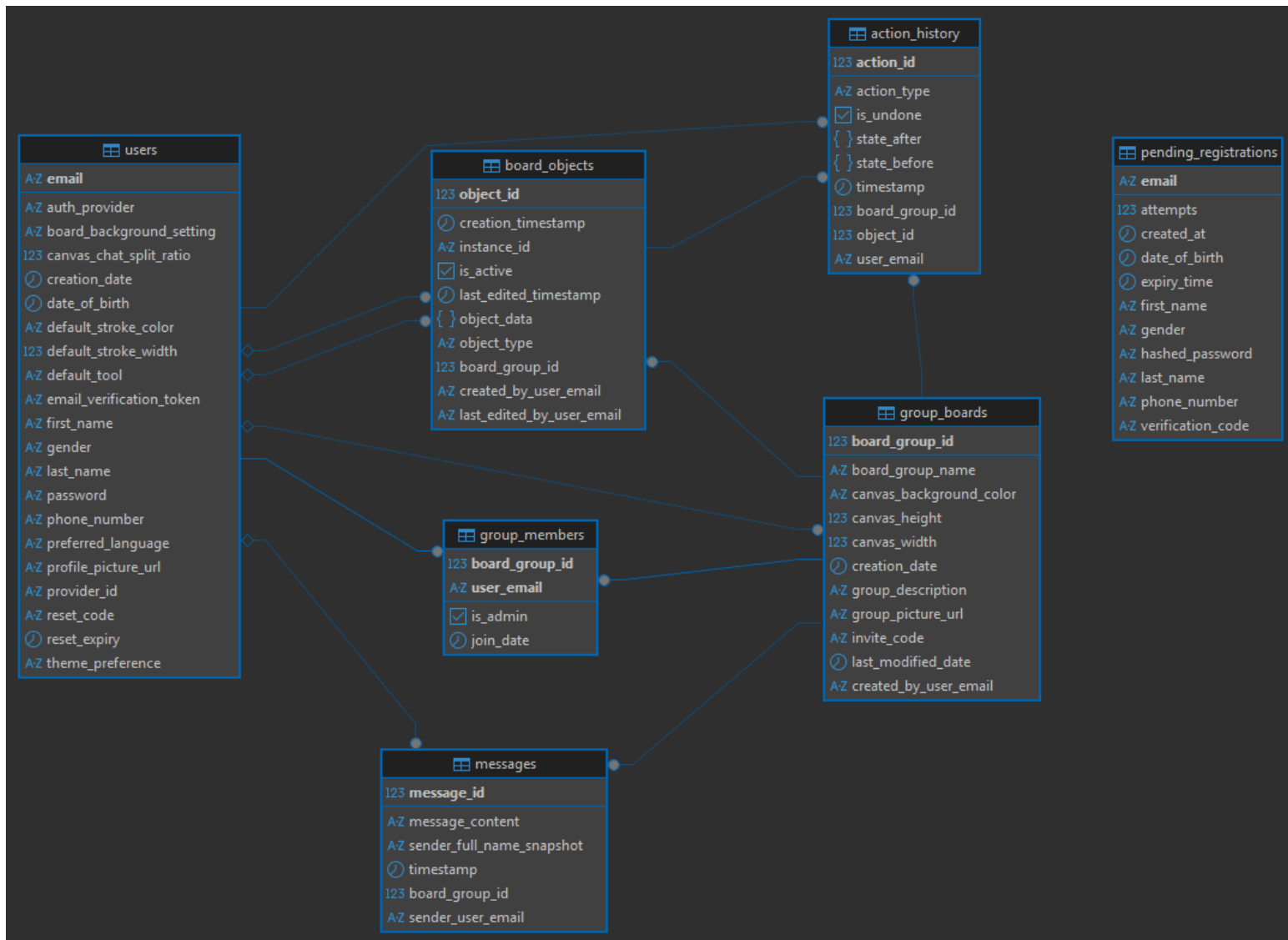
סכמת בסיס הנתונים

סכמת הנתונים מתרכזת סביב ישות ה-Users, אשר יכולים ליצור ולהצטרף ל-GroupBoards דרך טבלת החיבור GroupMembers. כל לוח מכיל מספר BoardObjects ו-Messages. טבלת ActionHistory עוקבת אחר כל השינויים כדי לאפשר פונקציונליות של undo / redo.

טבלאות מרכזיות:

שם טבלה	תיאור
users	מאחסנת את פרטי חשבונות המשתמשים, פרטי אימות, העדפות ונתוני פרופיל.
group_boards	מאחסנת מידע על הלוחות השיתופיים, כולל הגדרות, מטא-דאטה ומידות קנבס.
group_members	טבלת חיבור המנהלת את הקשר רבים לרבים בין משתמשים ללוחות, כולל הרשאות ניהול.
board_objects	מאחסנת את אלמנטי הציור הבודדים על גבי הלוחות. הנתונים הוויזואליים של כל אובייקט נשמרים בפורמט JSON.
messages	מאחסנת את הודעות הצ'אט הנשלחות בסביבת העבודה של כל לוח.
action_history	עוקבת אחר כל הפעולות של אובייקטים בלוח לצורך שחזור פעולות.
pending_registrations	אחסון זמני עבור הרשמות משתמשים הממתינות לאימות אימייל.

תרשים קשרי הגומלין בין הטבלאות:



נספח: הוראות התקנת המערכת

הפרויקט תומך בשני מצבי הרצה עיקריים:

- 1) סביבת פרודקשן (באמצעות Docker): הדרך המומלצת והקלה ביותר להרצת המערכת השלמה. כל השירותים (Frontend, Backend, בסיס הנתונים, Message Broker) רצים בקונטיינרים מבודדים.
 - 2) סביבת פיתוח (הרצה לוקאלית): מיועדת לפיתוח פעיל ודיבאגינג. בשיטה זו, ה-Frontend וה-Backend רצים ישירות על המחשב שלך (מתוך IDE או טרמינל), בעוד ששירותי התשתית (בסיס הנתונים וה-Message Broker) רצים ב-Docker.
- לפני שמתחילים יש להגדיר את שירותי צד שלישי האופציונליים כדי לאפשר את מלוא היכולת של המערכת.

1) הרצה בסביבת פרודקשן (Docker):

- שיטה זו מריצה את כל המערכת בפקודה אחת, אידיאלית לבדיקות או להרצה בסביבה המדמה פרודקשן.
- צור קובץ סביבה: העתק את קובץ הדוגמה כדי ליצור קובץ הגדרות עבור Docker. קובץ זה נמצא בתיקיית השורש של הפרויקט.

```
cp .env.example .env
```

- הגדר את משתני הסביבה:
 - פתח את קובץ ה-.env. שיצרת.
 - הדבק את המפתחות של Google ו-SendGrid (אופציונלי).
 - שנה אתה את הערך של JWT_SECRET_KEY למחרוזת אקראית וחזקה משלך.

- הרץ את המערכת: פקודה זו תבנה את האימג'ים של Frontend וה-Backend ותריץ את כל הקונטיינרים:

```
docker-compose up -build
```

- פתח את האתר: לאחר שהכל עלה, המערכת זמינה בכתובות הבאות:

○ Frontend (ממשק המשתמש): <http://localhost>

○ Backend API: <http://localhost:8080>

○ ActiveMQ Console: <http://localhost:8161> (שם משתמש

admin / admin).

(2) הרצה בסביבת פיתוח (לוקאלית):

שיטה זו מאפשרת לך לפתח ולדבאג את ה-Frontend וה-Backend תוך שימוש ב-Docker רק בשביל שירותי התשתית.

• הרץ את התשתית: הפעל רק את בסיס הנתונים ואת ה-Message

Broker באמצעות Docker. הדגל d- מריץ אותם ברקע:

```
docker-compose up -d postgres activemq
```

• הגדר והרץ את Backend:

○ פתח טרמינל חדש ונווט לתיקיית Backend.

○ צור קובץ סביבה ייעודי עבור Backend (זהו קובץ נפרד

מהקובץ שבתיקיית השורש):

```
cd backend
```

```
cp .env.example .env
```

○ פתח את הקובץ backend/.env והבדק בו את מפתחות ה-API של

Google ו-SendGrid (אופציונלי).

○ הרץ את שרת Backend באמצעות Gradle:

```
./gradlew bootRun
```

• הגדר והרץ את Frontend:

○ פתח טרמינל נוסף ונווט לתיקיית Frontend.

○ התקן את התלויות:

```
cd frontend
```

```
npm install
```

○ צור קובץ סביבה ייעודי עבור Frontend (זהו קובץ נפרד

מהקובץ שבתיקיית השורש):

```
cd frontend
```

```
cp .env.example .env
```

○ הרץ את שרת הפיתוח של Frontend:

`npm run dev`

(3) הפעלת שירותי צד שלישי (אופציונלי):

כדי להפעיל תכונות כמו התחברות עם גוגל ואימות באמצעות אימייל, עליך להנפיק מפתחות API מתאימים.

הגדרת התחברות עם גוגל (Google OAuth2):

תהליך זה יאפשר למשתמשים להירשם ולהתחבר באמצעות חשבון הגוגל שלהם.

- כנס ל-Google Cloud Console: נווט אל:

[/https://console.cloud.google.com](https://console.cloud.google.com)

- צור פרויקט חדש או בחר פרויקט קיים.

- נווט למסך ההגדרות: בתפריט הניווט, לך אל credentials ואז אל Apis & Service.

- צור Client ID חדש:

- לחץ על Create Credentials ובחר OAuth client ID.

- בסוג באפליקציה בחר Web application.

- תן שם כלשהו.

- הגדר Authorized redirect URIs:

- לחץ ADD URI והוסף את שתי הכתובות הבאות. שתיהן

נדרשות כדי לתמוך גם בסביבת הפיתוח וגם בסביבת

הפרודקשן:

<http://localhost:8080/login/oauth2/code/google>

<http://localhost/api/login/oauth2/code/google>

- העתק את המפתחות: לאחר יצירת ה-Client ID, גוגל יספק לך Client

Secret ID. העתק את שניהם.

- עדכן את קובץ הסביבה: הדבק את המפתחות שהעתקת למשתנים

GOOGLE_CLIENT_ID ו-GOOGLE_CLIENT_SECRET בקובץ

ה-`.env`. המתאים.

הערה: אם תשאיר משתנים אלה ריקים, אפשרות ההתחברות עם גוגל פשוט לא תופיע בממשק המשתמש, והמערכת תמשיך לעבוד כרגיל עם הרשמה רגילה.

הגדרת שירות אימיילים (SendGrid):

תהליך זה יאפשר למערכת לשלוח אימיילים לאימות משתמשים חדשים.

- צור חשבון בSendGrid: הירשם לשירות באתר:
[/https://sendgrid.com](https://sendgrid.com)
- הנפק מפתח API:
 - בתפריט הניווט לך אל API Keys ואז אל Settings.
 - לחץ על Create API Key.
 - תן שם למפתח ובחר בהרשאות Full Access.
 - העתק את המפתח שנוצר ושמור אותו במקום בטוח.
- עדכן את קובץ הסביבה: הדבק את המפתח שהעתקת למשתנה
SENDGRID_API_KEY בקובץ ה.env.