

פרויקט סיום בקורס 20503

סדנה בתכנות מתקדם בשפת Java

שם הפרויקט:

SynchBoard



מגיש: שגיא מנחם, 208645937

תאריך הגשה: 23/09/2025

1. מבוא:

הוא יישום Full-Stack המשמש כלוח ציור שיתופי, המועד לשיתוף פעולה ועבודה יזואלית בצוותים בזמן אמת. הישום מאפשר למספר משתמשים לציר, לשרטט ולתalking על גבי לוחות דיגיטליים משותפים, עם סנכרון מיידי של כל פעולה. הפלטפורמה מיעדת לצוותים, אנשי חינוך וארגוני הקיימים לגיל עבודה יזואלי אינטראקטיבי.

הפרויקט כולל מערכת מקיפה של אימות משתמשים, ניהול לוחות, צ'אט מובנה, כלי ציור וניהול הרשות לחברי הלוח. המערכת פותחה בטכנולוגיות Web מודרניות ומczyua 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 אימיל, טיפוס סיסמה והתחברות דרך OAuth2 (Google).
- ניהול לוחות מתקדמי: יצירה, הזמנת חברים ובקורת גישה מבוססת הרשות.
- כלים ציור עשירים: מברשות, צורות, קוים, טקסט, בחירת צבעים, הגדרות עובי קו ופונקציונליות Undo / Redo.
- צ'אט מובנה: מערכת צ'אט בזמן אמת המשולבת בכל לוח ומאפשרת תקשורת צוותית.
- עיצוב רספונסיבי ותמייה בריבוי שפות: תמייה מלאה במחשבים שולחניים ובטאבלטים, עם ממשק מלא באנגלית ובעברית (כולל תמייה RTL).
- ניהול קבצים: יכולות העלאת תמונות ללוח, תמונות פרופיל וייצוא של לוחות שהושלמו.

2. מדריך למשתמש:

תפקידי משתמשים:

למערכת SyncBoard שלושה סוגי משתמשים, שלכל אחד מהם הרשות שונות:

- **אורח:**

- יכול לגשת רק לעמוד ההתחברות והרשמה.
- אינו יכול לצפות בלוחות, להציג אליהם או להשתתף בסביבה העבודה השיתופית.
- חייב להשלים תהליך הרשמה (ואimoto אמייל) כדי לקבל גישה לאפליקציה.

- **משתמש רשום:**

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

- **מנהל לוח:**

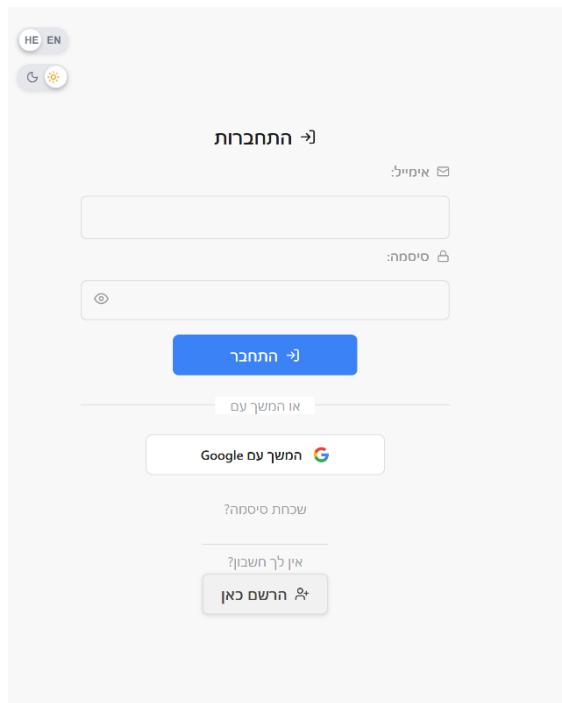
- בעל הרשות של משתמש רשום.
- בנוסף, יכול להזמין חברים חדשים ללוח, להסיר חברים קיימים, לקדם משתמשים אחרים למעמד מנהל, ולשנות את הגדרות הלוח (שם, תיאור, מידות קנבס וצבע רקע).

מדריך למשתמש

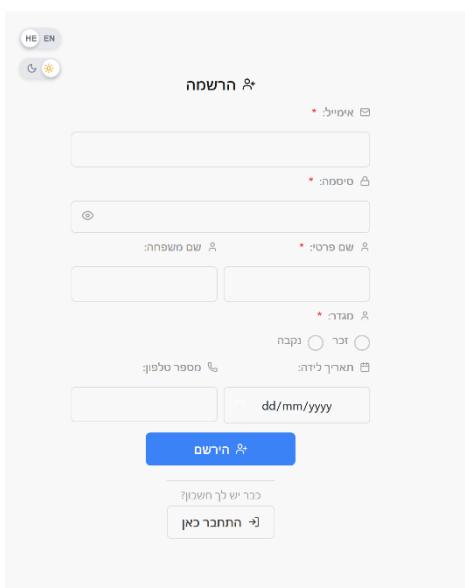
תהליכי עבודה מרכזים:

- הרשמה משתמש חדש:

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



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



מדריך למשתמש

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

ההרשמה

אימייל: mike.product@techcorp.com

סיסמה: *****

שם משפחה: Rodriguez

שם פרטי: Mike

מין: זכר

תאריך לידה: 05/00/0000

ההרשמה

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

Welcome to SyncBoard!

Verify Your Email Address

Thank you for registering with SyncBoard. To complete your registration, please enter the verification code below:

872810

Important:

- This code will expire in 15 minutes
- Enter the code exactly as shown above
- If you didn't request this, please ignore this email

אימות אימייל

ביטול

הזן קוד בן 6 ספרות

בדוק באימייל שלך את הקוד האמור

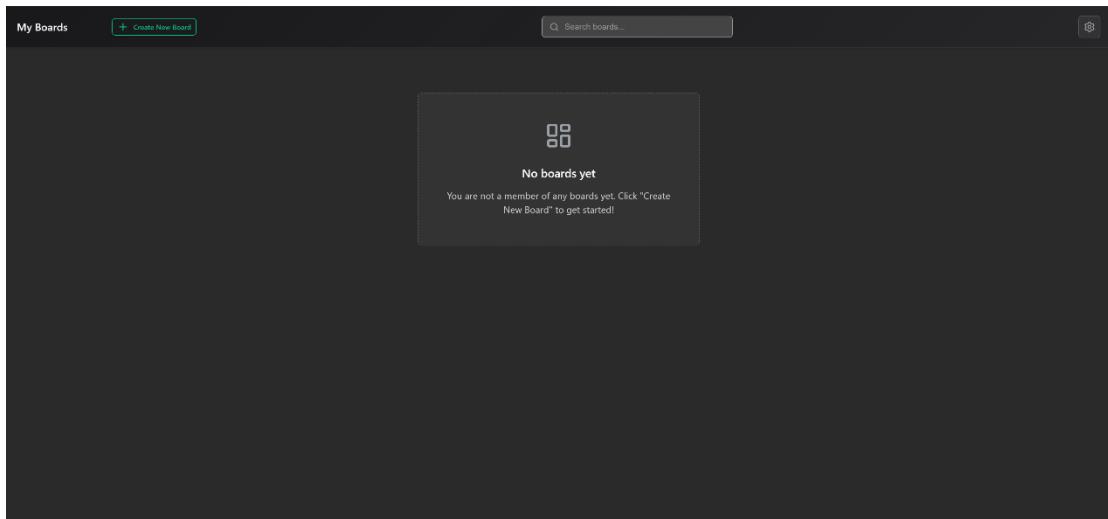
לא קיבלת את הקוד?

שלח קוד חדש

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

מדריך למשתמש

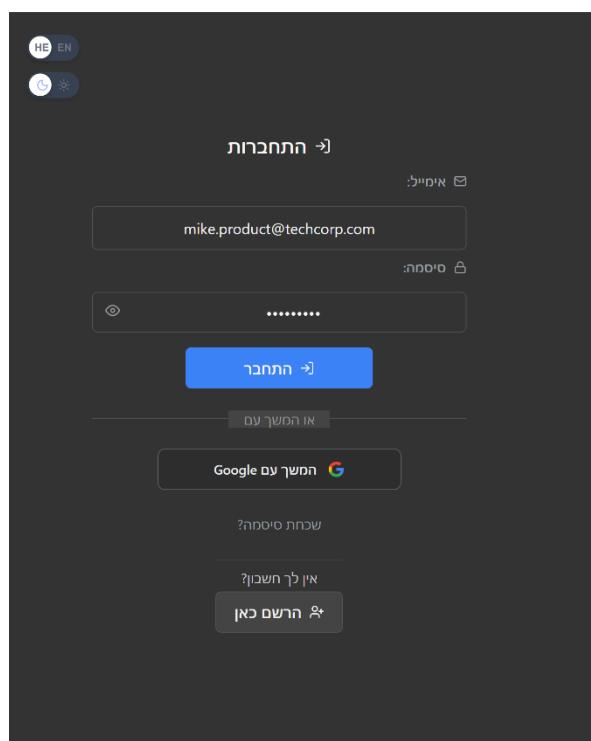
5) סיום והכוונה: לאחר אימות מוצלח, תועבר אוטומטית לעמוד רשימת הלוחות, שם תוכל להתחל ליצור לוחות חדשים.



- התחברות למערכת

תהליך כניסה למערכת עם חשבון קיימ.

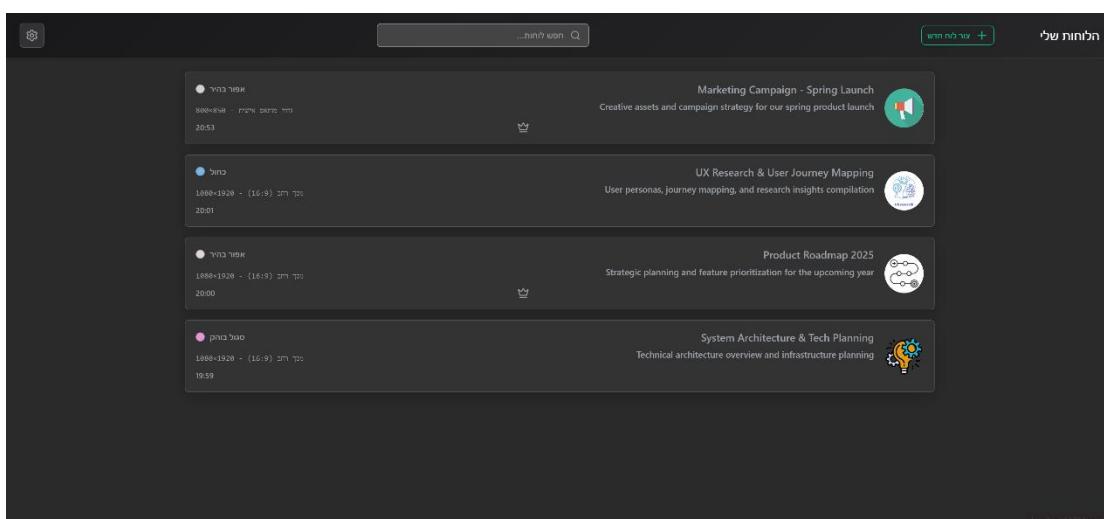
- 1) הזנת אימייל: בעמוד האימות, ודא שלשונית 'Login' פעליה והזן את כתובת האימייל הרשומה שלך.
- 2) הזנת סיסמה: הzin את הסיסמה בשדה המועד. ניתן להשתמש באיקון העין כדי להציג או להסתיר את הסיסמה.



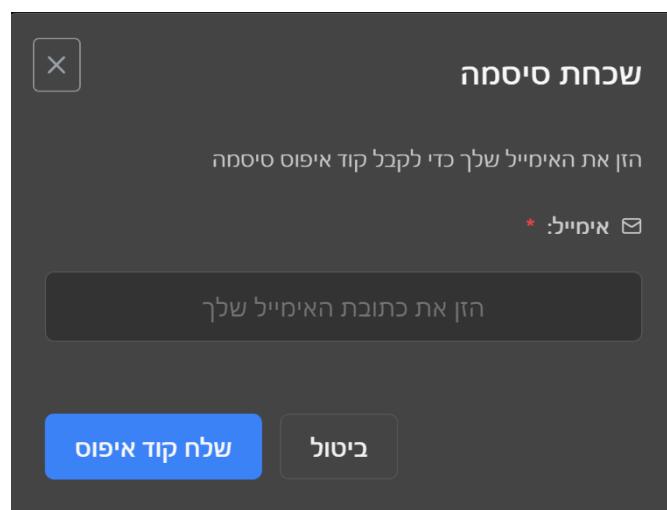
מדריך למשתמש

3) אימות: לחץ על כפתור 'התחבר' כדי להתחבר. לחלוין, אם נרשמת עם Google לחץ על כפתור 'המשך עם Google'.

4) הכוונה לעמוד הראשי: עם התחברות מוצלחת, תועבר אוטומטית לעמוד רשימת הלוחות שלך.



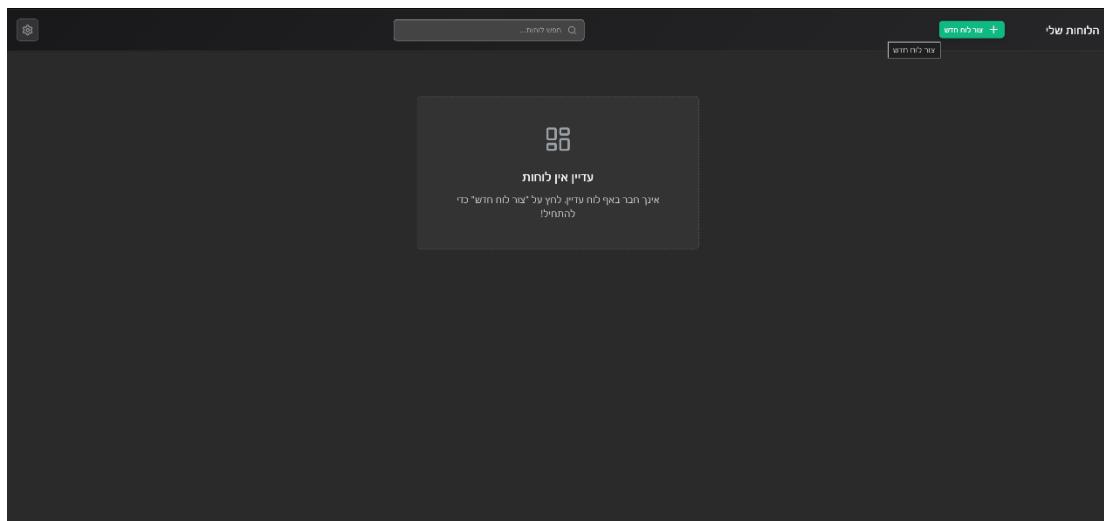
5) שכחת סיסמה: אם שכחת את הסיסמה, לחץ על הקישור 'שכחת סיסמה?' מתחת לתווך ההתחברות כדי להתחיל תהליך איפוס סיסמה באמצעות המail.



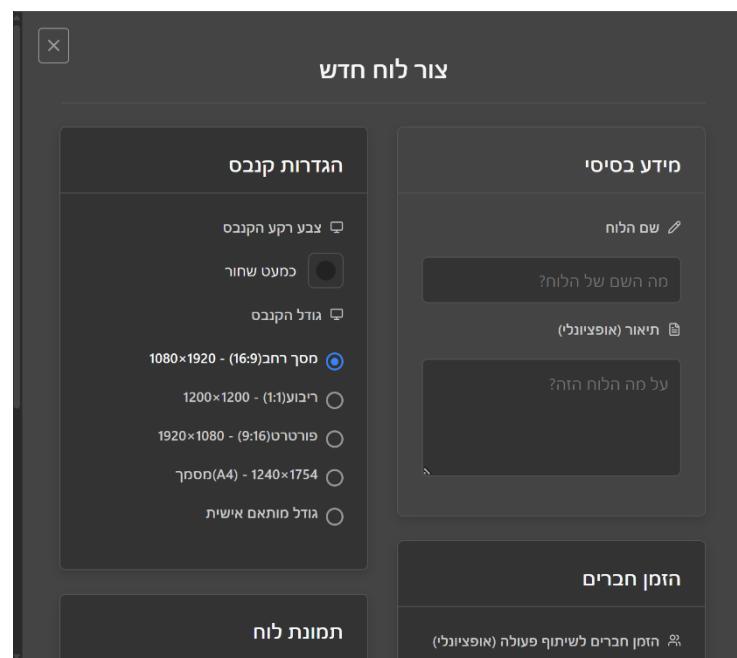
מדריך למשתמש

• יצירת לוח חדש

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

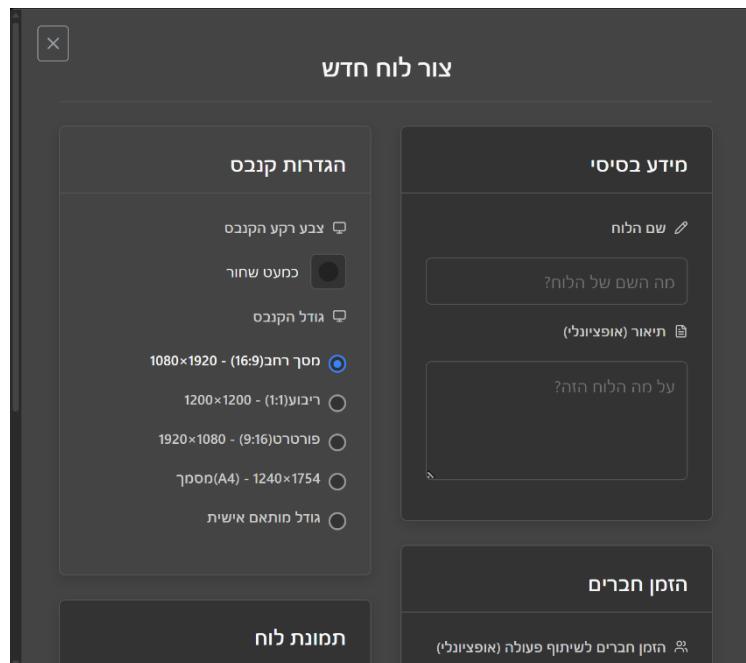


- 2) הגדרת שם ותיאור: בחלון שנפתח, הzin שם ללוח (3-100 תווים, חובה) ותיאור אופציונלי.

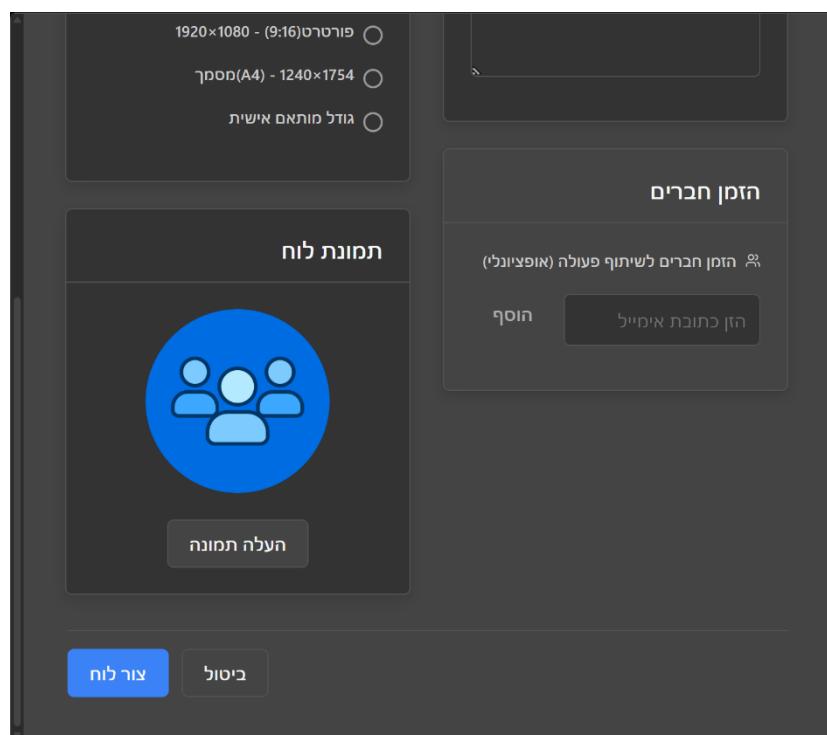


מדריך למשתמש

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



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

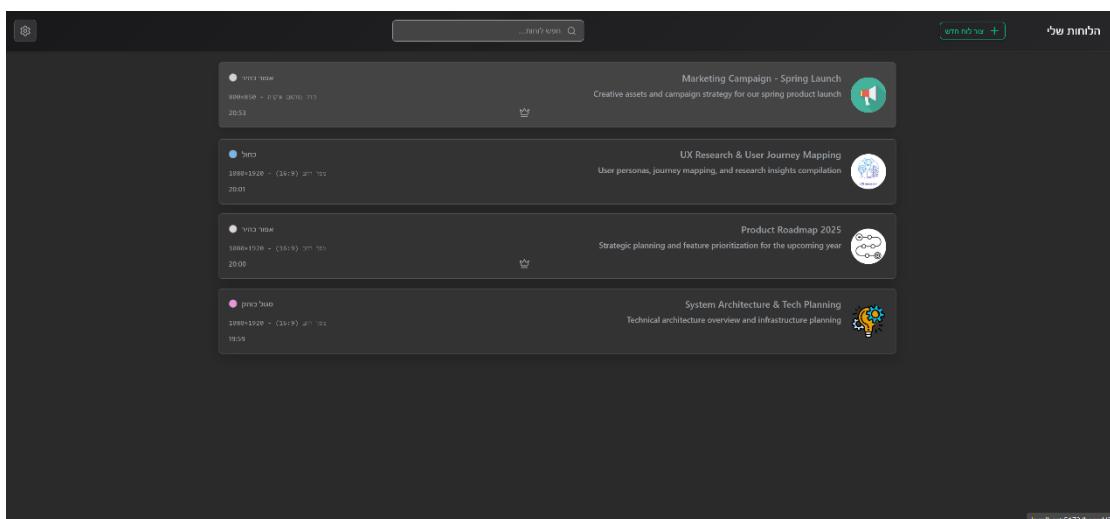


מדריך למשתמש

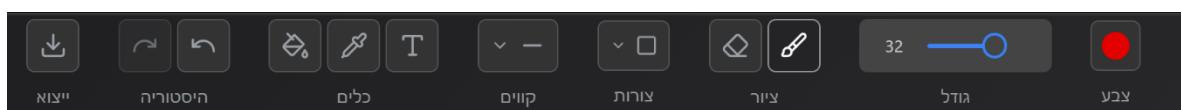
5) סיום יצירה: לחץ על הכפתור 'צור לוח' כדי לסיים את התהילה.
תועבר אוטומטית לSUBJECT הובודה של הלוח החדש.

- **עבודה על לוח שיתופי:**

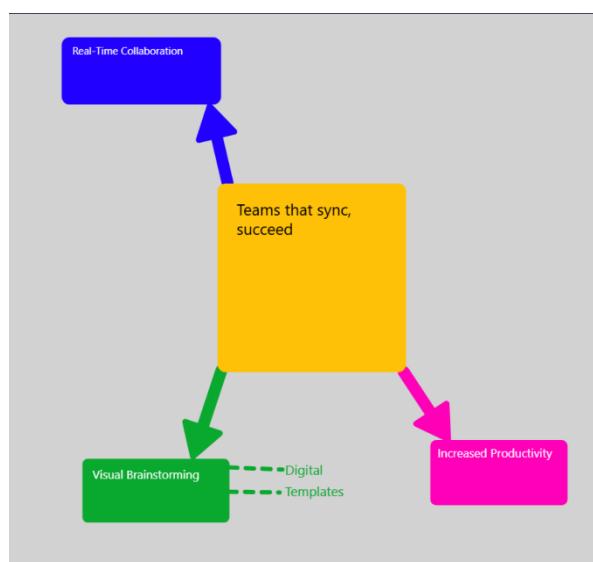
1) כניסה ללוח: לחץ על כרטיסיית הלוח הרצוי מרשימת הלוחות שלך.



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



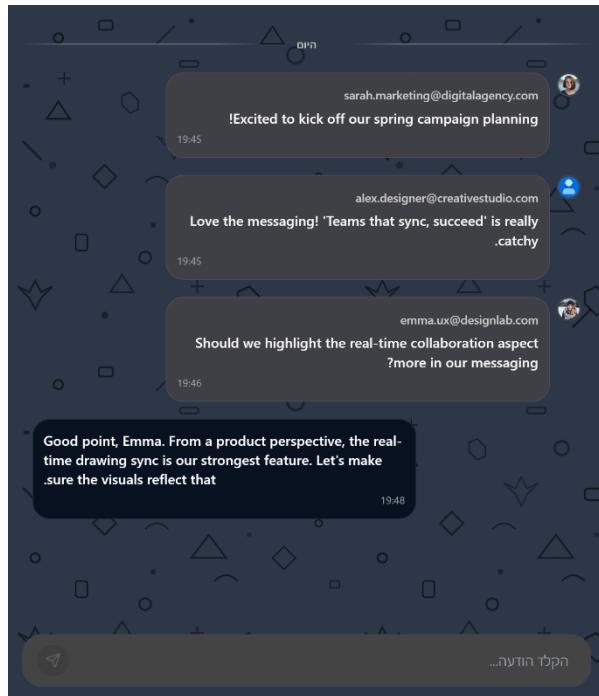
3) ציור על הקנבס: התחיל לצייר על הקנבס באמצעות לחיצה וגרירה של העכבר.



מדריך למשתמש

4) שימוש ביצ'אט: השתמש בחלונית היצ'אט לצד ימין כדי לתקשר עם

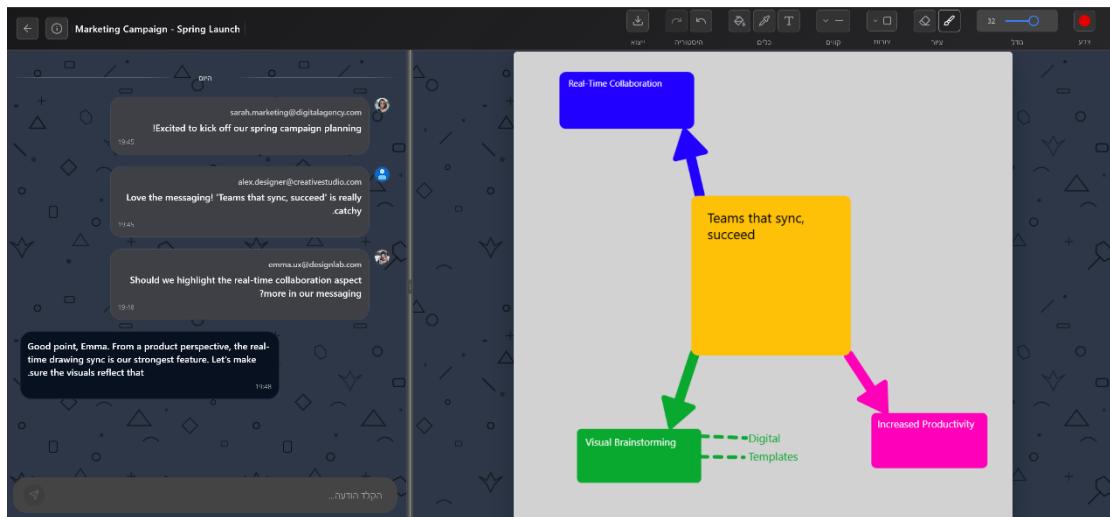
חברי הצוות בזמן אמת.



5) פעולות נוספות: השתמש בכפתורי Undo / Redo בסרגל הכלים

כדי לבטל או לשחזר פעולות. השתמש בכפתור ההורדה כדי ליצא

את מצב הלוח הנוכחי לקובץ תמונה.



3. תיעוד טכני:

תחום	טכנולוגיה	גרסת	מטרה
צד לקוח	React	19.1.1	ספריית UI ראשית לבניית ממשק משתמש אינטראקטיבי.
	TypeScript	5.9.2	פיתוח עם בטיחות טיפוסים לשיפור חווית הפיתוח.
	Vite	7.0.0	כלי בנייה ושרת פיתוח מודרני.
	React Router	7.6.3	ניהול ניתוב צד לקוח באפליקציית Single-Page.
	SCSS	1.91.0	פירה פרוessor LESS CSS עם תמייה במודולים לעיצוב קומפוננטות.
	@stomp/stompjs	7.1.1	תקשרות WebSocket באמצעות פרוטוקול STOMP.
	Axios	1.10.0	קlienט HTTP לתקשרות עם API RESTful.
	React i18next	15.6.0	ספירה לניהול תרגומים (אנגלית / עברית) ותמייה ב-RTL.
	Java	24	שפת התכונות הראשית לפיתוח צד שרת.
צד שרת	Spring Boot	3.5.5	Framework ראשי לשימוש, מספק הזרקת תלויות ותוכנות Production-ready.
	Spring Security	כלול	ניהול אבטחה, אימות, הרשאות, טוקני JWT ואינטגרציה OAuth2.
	Spring WebSocket	כלול	תמייה ב-WebSocket עם STOMP בזמן אמיתי.
	Spring Data JPA	כלול	שכבת גישה לנוטונים עם תבנית Repository ושאילות אוטומטיות.
	Hibernate	כלול	מיפוי אובייקטיבי רלציוני לניהול ישויות בסיס הנתונים.
	SendGrid	4.10.3	אינטרגרציה עם שירות שליחת אימיילים לאימות ואייפוס סיסמה.
	JSON Web Tokens	0.13.0	יצירה ואיימות של טוקני אימות Stateless.
	Gradle	כלול	כלי אוטומציה לבנייה וניהול תלויות.
	PostgreSQL	17	בסיס נתונים רלציוני ראשי לאחסון נתונים משתמשים, לוחות, אובייקטים ועוד.
בסיס נתונים	Docker	Latest	פלטפורמת קונטינריזציה להרצה עקבית בסביבות שונות.
	Docker Compose	Latest	תזמור של מספר קונטינרים.
	Nginx	Latest	שרת Web להגשת קבצים סטטיים של Frontend.
Deployment			

מודול WebSocket Message Broker airyuim בזמן אמת.	2.31.0	ActiveMQ Artemis	
---	--------	------------------	--

טכנולוגיות בשימוש:

ארquitektורה:

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

סכמת בסיס הנתונים

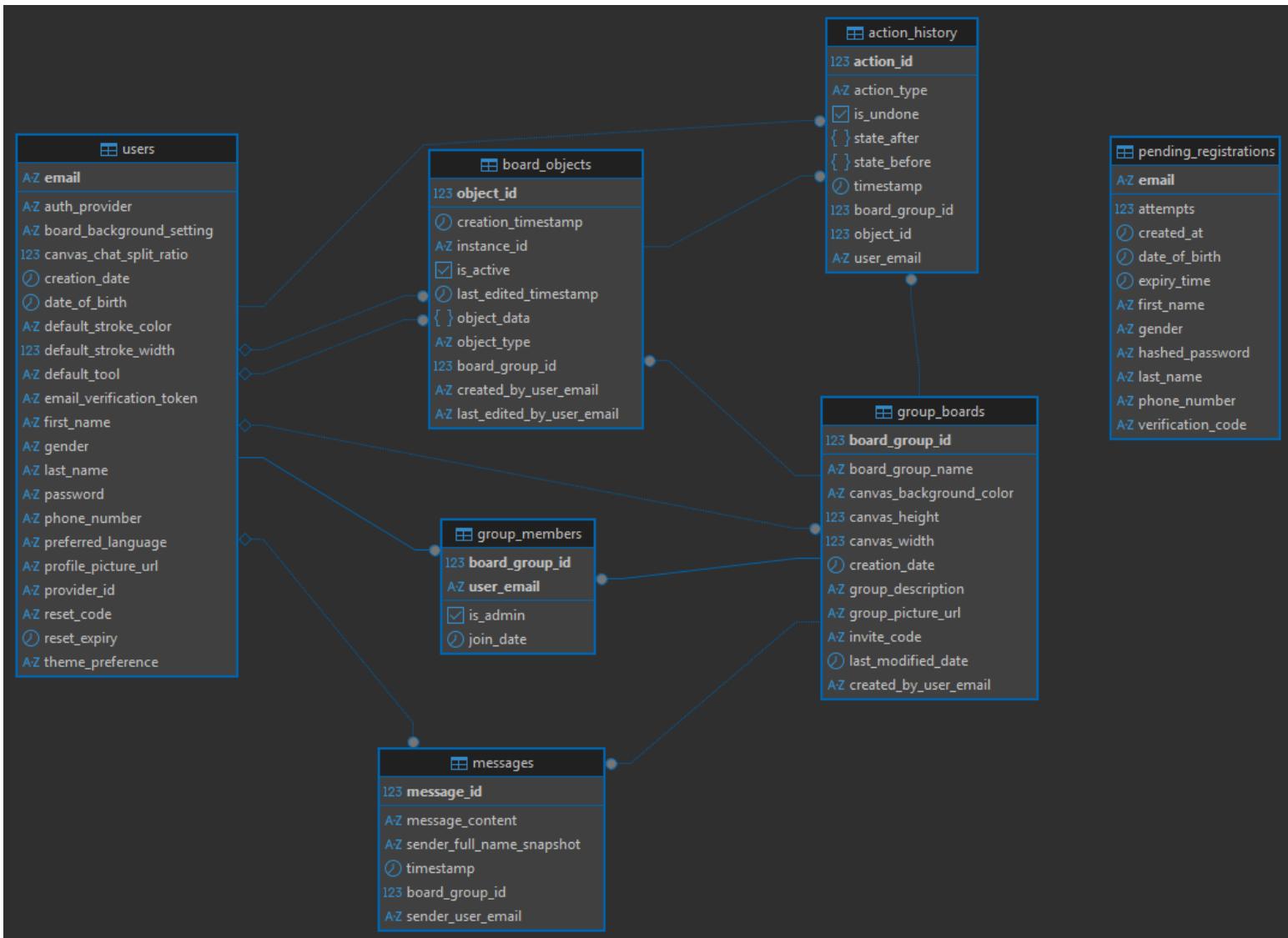
סכמת הנתונים מתרכזת סביב ישות Users, אשר יכולים ליצור ולהציגן GroupBoards דרך טבלת החיבור GroupMembers. כל לוח מכיל מספר ActionHistory ועקבות אחר כל השינויים Messages וBoardObjects כדי לאפשר פונקציונליות של undo / redo .

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-דата ומידות קבבם.
group_members	טבלת חיבור המנהלת את הקשר רבים 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	עקבות אחר כל הפעולות של אובייקטים בלוח לצורך שחזור פעולות.
pending_registrations	אחסון זמני עבור הרשומות המשתמשים המתיננות לאימайл.

מדריך למשתמש

תרשים קשרי הגומלין בין הטבלאות:



בוסף: הוראות התקנת המערכת

הפרויקט תומך בשני מצבים רצאה עיקריים:

- 1) סביבת פרודקشن (באמצעות Docker): הדרך המומלצת והקלת ביותר להרכבת המערכת המלאה. כל השירותים (Frontend, Backend, הנטונים, Message Broker) רצים בקונטינרדים מבודדים.
- 2) סביבת פיתוח (רצאה לokaלית): מיועדת לפיתוח פעיל ודיבאגינג. בשיטה זו, h-Backend ו-h-Frontend רצים יישור על המחשב שלך (מתוך IDE או טרמינל), בעוד ששירותי התשתייה (בבסיס הנטונים והזבקר) (Message Broker) רצים ב-Docker.

לפניהם של מתחילה יש להגדיר את שירותים צד שלישי האופציונליים כדי לאפשר את מלאה היכולת של המערכת.

1) רצאה בסביבת פרודקشن (Docker):

- שיטה זו מרים את כל המערכת בפקודה אחת, אידיאלית לבדיקות או להרכבה בסביבה המדממת פרודקشن.
- 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 SendGrid ו-Google (אופציונלי).
 - שנה אתה את הערך של JWT_SECRET_KEY למחרוזת אקראיית וחזקת משלך.
 - הרץ את המערכת: פקודה זו תבנה את האימג'ים של Frontend ו-Backend ותריץ את כל הקונטינרים:
docker-compose up -build
 - פתח את האתר: לאחר שהכל עלה, המערכת זמינה בכתובות הבאות:
 - <http://localhost> (ממשק המשתמש): Frontend
 - <http://localhost:8080>: Backend API

- (שם משתמש <http://localhost:8161> : ActiveMQ Console) וסיסמה (admin / admin).
- (2) הרצה בסביבת פיתוח (локאלית):

שיטה זו מאפשרת לך לפתח ולדבаг את ה-Backend וה-Frontend תוך שימוש ב Docker רק בשביל השירותי התשתיות.

- הרץ את התשתיות: הפעיל רק את בסיס הנתונים ואת ה-Message Broker באמצעות Docker. הדגל -d מרים 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-APIים וwhether he has the correct configuration for SendGrid and Google (optional).
- הרץ את שרת Backend באמצעות Gradle:
.gradlew bootRun
- הגדר והרץ את Frontend:
 - לפתוח 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 שירות צד שלישי (אופציוני):

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

הגדרת התחברות עם גוגל (Google OAuth2):

תהליך זה מאפשר למשתמשים להירשם ולהתחבר באמצעות חשבון הגוגול שלהם.

- כניסה ל-Google Cloud Console: נווט אל:
<https://console.cloud.google.com>
- >Create Credentials נווט למסך ההגדרות: בתפריט הניווט, לך אל credentials ואז אל APIs & Services.
- Create Client ID נווט לחץ על Create Credentials ובחירה Client ID ובחירה Web application.
- בסוג אפליקציה בחר Web application.
- תן שם כלשהו.
- הגדיר URLs Allowed redirect URLs: נווט לחץ ADD והוסיף את שתי הכתובות הבאות. שתיהן נדרשות כדי לתמוך גם בסביבת הפיתוח וגם בסביבת הprdוקשן:
 - localhost:8080/login/oauth2/code/google
 - http://localhost/api/login/oauth2/code/google

- העתק את המפתחות: לאחר יצירת Client ID, גוגל יספק לך Client Secret ID. העתק את שנייהם.
- עדכן את קובץ הסביבה: הדבק את המפתחות שהעתק למשתנים GOOGLE_CLIENT_SECRET ו-GOOGLE_CLIENT_ID בקובץ config.json. המתאים.

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

הגדרת שירות אימיילים (SendGrid):

זהה להיליך זה יאפשר למערכת לשלוח אימיילים לאיומות משתמשים חדשים.

- צור חשבון בSendGrid: הירשם לשירות באתר:

[/https://sendgrid.com](https://sendgrid.com)

- הנפק מפתח API:

.Settings API Keys ואז אל

.Create API Key לחץ על

.Full Access תן שם למפתח ובחר בהרשאות.

העתק את המפתח שנוצר ושמור אותו במקום בטוח.

- עדכן את קובץ הסביבה: הדבק את המפתח שהעתקת למשתנה

.enrcd SENDGRID_API_KEY בקובץ זהה.