

## תוכן עניינים

3	1. הגדרת דרישות ותיאור כללי
3	1.1 תיאור כללי
3	1.1.1 מטרות המערכת
3	1.1.2 היקף הפרויקט
3	1.1.3 מבנה וארגון
3	1.1.4 משימות המערכת
4	1.2 תיאור חומרת המערכת
4	1.2.1 כללי
4	1.2.2 מרכיבי המערכת
4	1.3 תיאור תוכנת המערכת
4	1.3.1 כללי
4	1.3.2 תיאור ה-SYSTEM
5	1.3.3 כלי התוכנה לפיתוח המערכת
5	1.4 תיאור פונקציות המערכת
6	1.5 זרימת המידע במערכת
6	1.5.1 כללי
6	1.5.2 תרשימים
9	1.5.3 מילון מונחים
9	phpMyAdmin
10	2. ממשקים חיצוניים
10	3. ממשק אדם מכונה
10	3.1 כללי
11	3.2 תיאור מסכים
12	3.3 תיאור מסכים והסבר
23	4. מבנה נתונים וארגון הקבצים
23	4.1 ארגון קבצים
23	4.1.1 כללי
25	4.2 מבנה נתונים
30	5. תכנון
30	5.1 כללי ועקרונות התכנות (5.2)
31	5.3 תיאור אלגוריתמים

34	6. מה הקנה הפרויקט.....
34	7. ביבליוגרפיה.....
34	7.1 ספרות.....
34	7.2 אתרי אינטרנט.....
35	8. נספחים.....
35	8.1 הוראות התקנה.....
35	8.2 מדריך למשתמש.....
39	8.3 קוד מקור מתועד.....
39	8.4 שונות.....
39	9. דרישות מערכת ופונקציונליות.....
39	9.1 דרישות לא פונקציונליות ודרישות מערכות.....
40	9.2 דרישות פונקציונליות.....
41	9.3 חלופות ארכיטקטוניות.....
41	9.3.1 ברמת המערכת.....
44	9.3.2 ברמת מכלולי התוכנה:.....
45	10. טופולגיית הפתרון הנבחר.....
46	11. ארכיטקטורה נבחרת.....
46	11.1 ראה סעיף 1.5 תרשימי מערכת.....
46	11.1.1 Data Flow.....
47	11.2 חלופות שפת מימוש.....
48	12. חלופות אבטחתיות והגנה.....
48	12.1 פירוט בדירות תוכנה ואופן ביצוען DOC-STP.....

## 1. הגדרת דרישות ותאור כללי

### 1.1 תיאור כללי

#### 1.1.1 מטרת המערכת

כיום, כמעט בכל ארגון, קטן עד גדול, ישנן משימות רבות לביצוע על ידי עובדים שונים.

ההנחה היא כי ניהול יעיל של תהליכים בארגון יכול לשפר את התחרות בין הארגונים, ואת יעילות ביצוע המשימות והתהליכים. בארגונים רבים, תהליכים ומשימות אלו לא מתועדים כראוי, ואפילו מתבצעים בעל פה, והעובדים אינם מכירים היטב את המשימות שיש לבצע. בנוסף, המנהל אינו מודע לקצב ביצוע המשימות ב-Real-time, ובעצם לא נמצא במעקב אחר המשימות והפרויקטים שבאחריות המחלקות השונות בארגון. וזוהי הבעיה הגדולה של המנהלים בארגון.

לכן, הקמנו את Elementech - מערכת מידע משולבת לניהול ושליטה אפקטיביים של פרויקטים ומשימות בתוך הארגון. בעזרת מערכת זו, מנהלי הארגון יכולים להקצות פרויקטים למחלקות מסוימות ולעובדים מסוימים, וכמו כן להתעדכן בסטטוס משימות ופרויקטים בארגון בזמן אמת. מצב זה מאפשר למנהל לתעדף משימות ופרויקטים, לצפות בנתוני ביצוע משימות ופרויקטים ולהיות בשליטה מתמדת על התהליכים בארגון.

המערכת מושתתת על בסיס נתונים הכולל: משתמשי המערכת, קבוצות משתמשים (מחלקות), פרויקטים, משימות, וסטטוסים.

#### 1.1.2 היקף הפרויקט

700 שעות לזוג.

#### 1.1.3 מבנה וארגון

המערכת מתבססת על תבנית העיצוב של שלושת השכבות, שכבת הנתונים, שכבת היישום ושכבת התצוגה – בשפה המקצועית MVC (Model – View – Controller) שעליה נרחיב בהמשך הספר.

קיים בסיס נתונים אשר משמש לשליפת נתונים, הצגתם, ולידציה ועוד.

המערכת כתובה ב-JavaScript ו-PHP, עושה שימוש בטכנולוגיית Ajax, המחזירה מידע מהשרת מבלי לרענן את הדף ולהריץ יישומים ברקע, כמו כן מתבצע שימוש בממשק משתמש המבוסס JavaScript – ששמו W2UI.

#### 1.1.4 משימות המערכת

- ניהול והצגת תקציר ממסד הנתונים, על ידי Dashboard – שמציג סך פרויקטים, משימות ופעילות משתמשים, בחלק הימני ב-Dashboard מופיעות הודעות ממשתמשי המערכת ע"מ להישאר בסנכרון מלא.
- צפייה ומעקב אחר פרויקטים, משימות ופעילויות בארגון.
- הקצאת משימות למחלקות שונות.
- ניהול משתמשים.
- ניהול פרויקטים.
- ניהול משימות ופעילויות במערכת.

## **1.2 תיאור חומרת המערכת**

### 1.2.1 כללי

- מחשב.
- מקלדת.
- עכבר.

### 1.2.2 מרכיבי המערכת

- שרת אחסון.
- רשת.
- חיבור לאינטרנט.

## **1.3 תיאור תוכנת המערכת**

### 1.3.1 כללי

המערכת תציג בפני המנהל את סטטוס הפרויקטים, המשימות והפעילויות בארגון בתצורת Dashboard נוחה וידידותית למשתמש. בנוסף, יוצגו באותו המסך גם הפעילויות של המנהל וגם הפעילויות של שאר המשתמשים.

קיים מסך לניהול פרויקטים, שבו ניתן להוסיף, למחוק, ולעדכן פרויקטים במערכת.

קיים מסך לניהול משימות, שבו ניתן להוסיף, למחוק, לעדכן משימות במערכת.

כמו כן, קיים מסך שבו ניתן לצפות במשימות השונות והתקדמותן, לאיזה פרויקט מקושרת משימה, מה העדיפות שלה על פי המשתמש, תאריכי התחלה וסיום, ולמי היא מוקצית.

ניתן להגדיר, לעדכן, ולמחוק משתמשים וקבוצת משתמשים שעל פי כל קבוצה ניתן לחלק משימות ופרויקטים שונים במערכת.

### 1.3.2 תיאור ה-SYSTEM

המערכת תציג בפני המנהל את סטטוס הפרויקטים, המשימות והפעילויות בארגון בתצורת Dashboard נוחה וידידותית למשתמש. בנוסף, יוצגו באותו המסך גם הפעילויות של המנהל וגם הפעילויות של שאר המשתמשים.

קיים מסך לניהול פרויקטים, שבו ניתן להוסיף, למחוק, ולעדכן פרויקטים במערכת.

קיים מסך לניהול משימות, שבו ניתן להוסיף, למחוק, לעדכן משימות במערכת.

כמו כן, קיים מסך שבו ניתן לצפות במשימות השונות והתקדמותן, לאיזה פרויקט מקושרת משימה, מה העדיפות שלה על פי המשתמש, תאריכי התחלה וסיום, ולמי היא מוקצית.

ניתן להגדיר, לעדכן, ולמחוק משתמשים וקבוצת משתמשים שעל פי כל קבוצה ניתן לחלק משימות ופרויקטים שונים במערכת.

### 1.3.3 כלי התוכנה לפיתוח המערכת

- Visual Studio / Visual Studio Code
- Ajax
- jQuery
- CSS
- W2UI – ממשק משתמש המבוסס JavaScript

### 1.4 תיאור פונקציות המערכת

פונקציות שלילפה, עדכון, מחיקה, הוספה, ושמירה מבסיס הנתונים.

#### ברמת המשתמש :

- התחברות למערכת.
- הרשמה למערכת.
- חיפוש משתמש.
- עדכון פרטי משתמש.
- יצירת משתמש חדש.
- וידוא שהמשתמש אינו קיים במערכת.
- וידוא מייל שאינו קיים במערכת.

#### ברמת הפרויקטים \ משימות \ פעילויות :

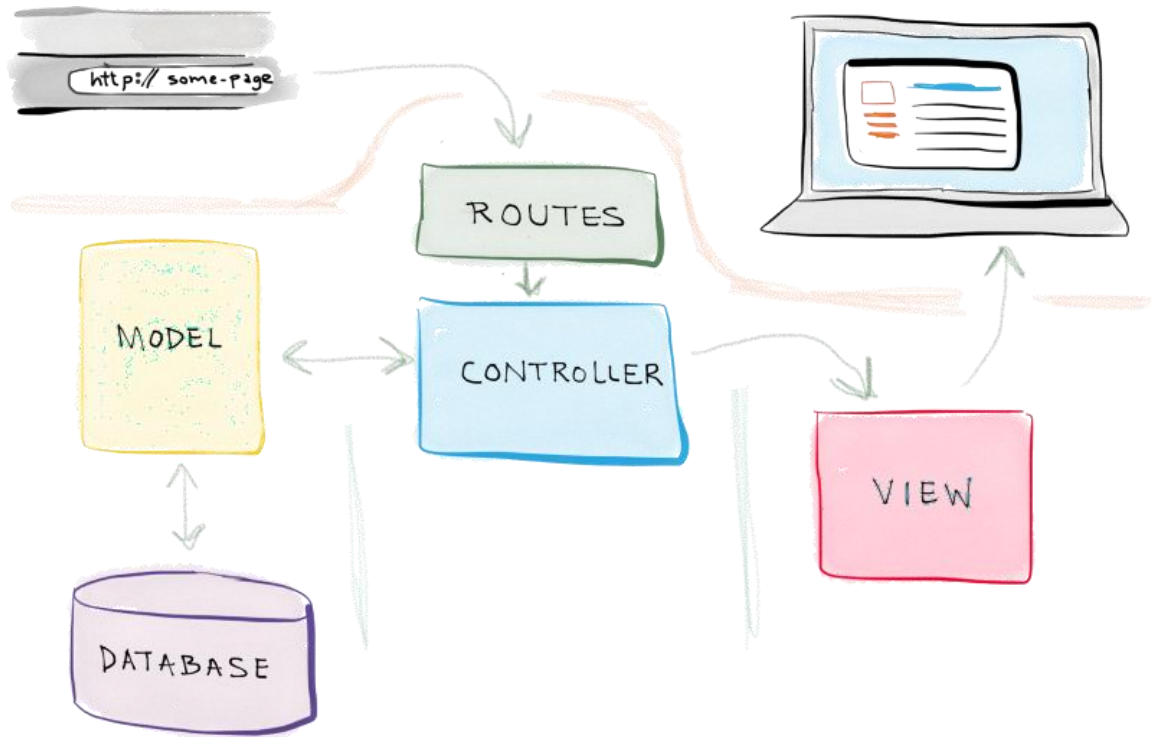
- חיפוש פרויקט \ משימה \ פעילות.
- עדכון נתוני פרויקט \ שם משימה.
- חיפוש פעילות על פי משימה.
- מחיקת פרויקט \ משימה \ פעילות.
- חיפוש משימה על פי פרויקט ומשתמש.
- שמירת פרויקט \ משימה \ פעילות.
- הוספת התקדמות משתמש.
- הצגת התקדמות משתמש.
- עריכת נתוני פרויקט \ משימה \ פעילות.

## 1.5 זרימת המידע במערכת

### 1.5.1 כללי

זרימת המידע במערכת מתבצעת על פי תבנית עיצוב MVC (Model-View-Controller) שלוש השכבות, התבנית מאפשרת חלוקה בין שלושה חלקים: מודל, תצוגה ובקר. בדרך זו התלות ההדדית בין ממשק המשתמש לשאר חלקי התוכנה פוחתת ואת החלקים השונים ניתן לפתח באופן בלתי תלוי. בנוסף קל יותר לתחזק את התוכנה ולבצע שימוש חוזר.

### 1.5.2 תרשימים



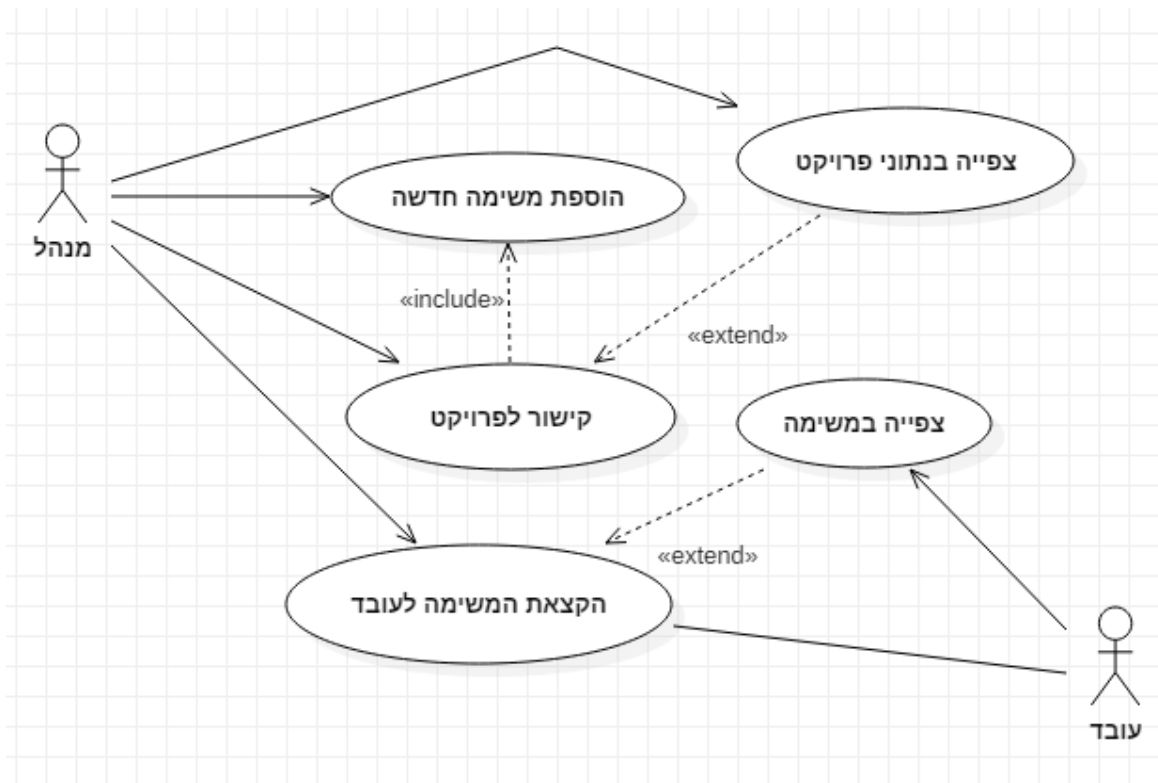
Model - מטפל בקריאה ובכתיבת המידע למסד הנתונים ובכל הפעולות שקשורות לליבה העסקית של האפליקציה: חישובים למיניהם, אבטחה וסיסמאות וכל מה שהוא לא תצוגה.

View - כל מה שקשור לתצוגה, ה-view הוא זה שמדפיס את ה HTML - ומחליט איפה למקם את המשתנים.

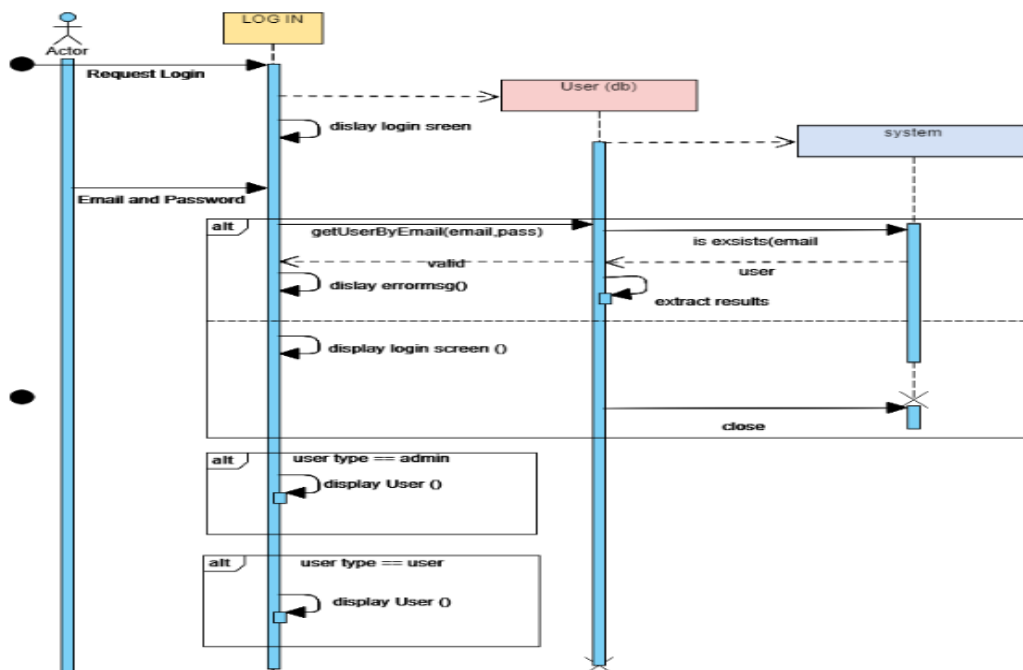
Controller - זה שמתווך בין השניים.

## תרשימי המערכת

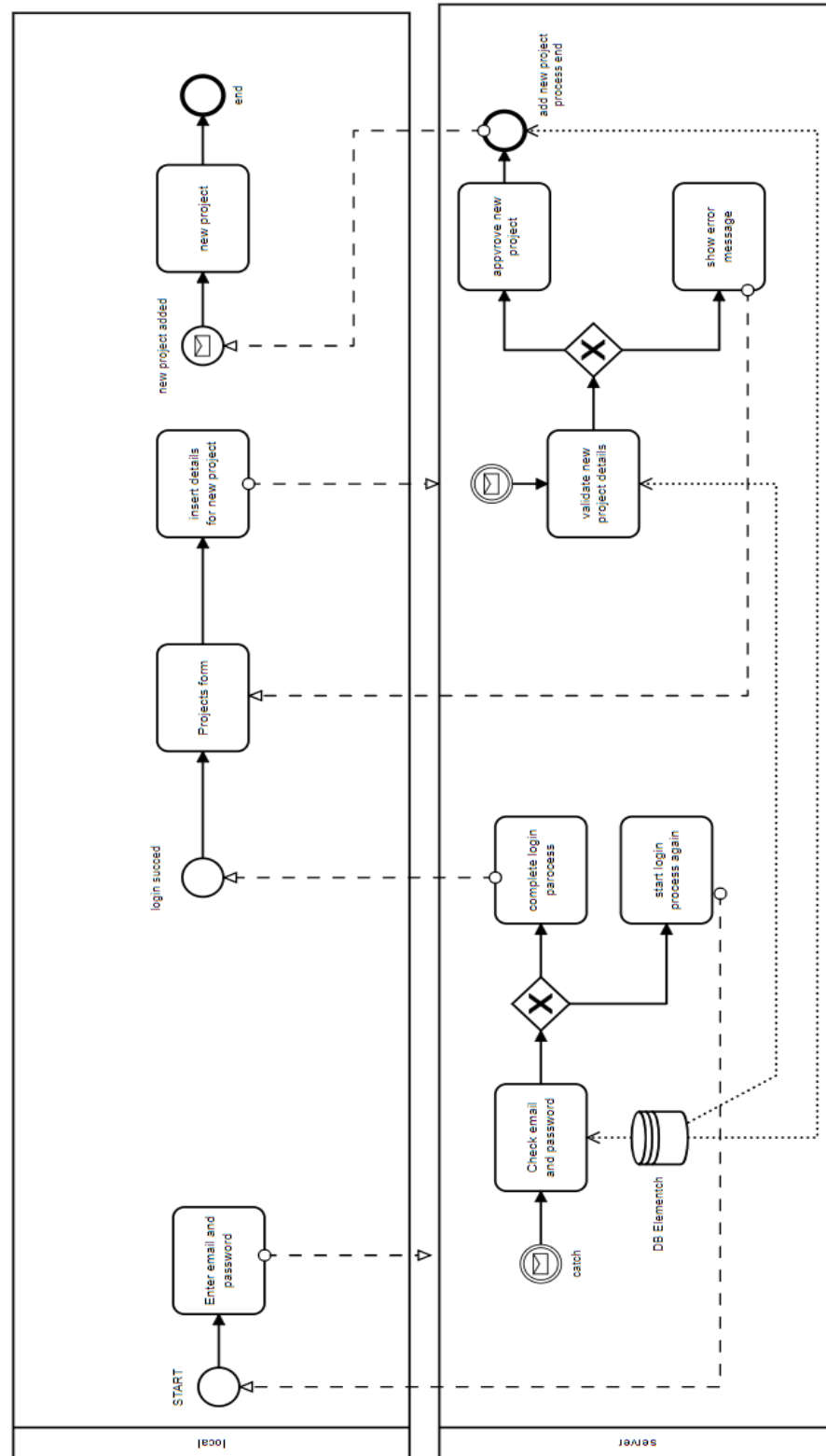
### Use Case - תהליך הוספת משימה חדשה עבור פרויקט והקצאתה למשתמש.



### Sequence Diagram - תהליך כניסה למערכת



# Business Process Management - תהליך בנייה למערכת והוספת פרויקט חדש





## 1.5.3 מילון מונחים

הגדרה	פירוש
XAMPP	תוכנה מאפשרת ליצור אפליקציות WEB עם שרת Apache תומכת בשפת PHP ובסיס נתונים MySQL
phpMyAdmin	זהו ממשק גרפי למשתמש (GUI) אשר משמש לניהול המידע בתוך מסד הנתונים MySQL
PHP	שפת סקריפט שרצה בצד השרת, פיתוח ישראלי
Java Script	היא שפת תכנות דינמית מונחית - עצמים המותאמת לשילוב באתרי אינטרנט ורצה על ידי דפדפן האינטרנט בצד הלקוח.
OOP	תכנות מונחה עצמים-Object Oriented Programming.
MVC	Model-View-Controller תבנית עיצוב בהנדסת תוכנה המשמשת להפשטת יישום כלשהו
Admin	מנהל המערכת. יכול לגשת לניהול מאגר השאלות על פי סיסמה אישית
Web	שיטת תקשורת בין רכיבי תוכנה ברשת מחשבים.
GUI	הוא ממשק משתמש לתוכנה או לאתר אינטרנט, המבוסס על עיצוב גרפי של המסך המוצג למשתמש
Design pattern	תבניות עיצוב הן פתרון כללי לבעיה שכיחה בעיצוב תוכנה. תבנית עיצוב אינה עיצוב סופי שניתן להעבירו הישר לקוד, אלא תיאור או תבנית לדרך לפתרון בעיה.

## 2. ממשקים חיצוניים

לא בוצע שימוש בממשקים חיצוניים במערכת.

## 3. ממשק אדם מכונה

### 3.1 כללי

#### Human-Machine Interface

למערכת קיים ממשק גרפי ידידותי למשתמש (GUI), מבנה המסכים והתצוגות עקבי במסכים המשתמשים בין היתר לעריכה, לצורך העניין מסך פרויקטים \ משימות \ משתמשים בעלי אותם מקשי עריכה \ הוספה \ עדכון ומופיעים בכל המסכים באותו המקום במסך, בנוסף יש שימוש נכון וחכם בצבעים, לדוגמא, פקד מחיקת המשתמש מופיע בצבע אדום, כדי שהמשתמש ויזואלית יבחין כי המקש הינו **מחיקה**, לאחר לחיצה על מחיקת משתמש, תקפוץ הודעת חיווי למשתמש, בחלונית קטנה, 'האם אתה בטוח שתרצה למחוק?'. כל חלון עומד 'מול' משימה ראשית, ניהול משתמשים \ פרויקטים \ משימות.

פעולות נוספות במסכים, כמו הוספה, מידע, חיפוש, ריענון דף, מופיעים עם אייקון מתאים המנסה לתאר את הפעולה בעת לחיצה על הפקד. בעת לחיצה על 'הוספה' לצורך העניין, יעלה מסך יותר קטן למשתמש כאשר כל שאר המסך אינו בפוקוס, והמשתמש יוכל להקיש את הנתונים הרלוונטיים, עבור כל שדה מתאים תופיע רשימת בחירה שממנה יוכל המשתמש לבחור מתוך בסיס הנתונים את הערך המתאים, על יד כל שדה שהוא שדה חובה תופיע '\*', לפני כתיבת הערך בשדה, השדה יציג את הפורמט המתאים לכתיבת שדה זה, לצורך הדוגמא בעת כתיבת תאריך לידה תפתח חלונית Calendar והפורמט כתיבה הנכון d-m-yyyy וזאת כדי לתת למשתמש את כל האינפורמציה לשימוש נכון ללא שגיאות במערכת.

פעולות שאינן "פעילות" במצבי חלון מסוימים יוצגו בצורה "חיוורת" ותחסם הפעלתן. ניסיון הפעלה יגרום להצגת הודעת חיווי מתאימה בחלון "הודעות מערכת".  
לאורך כל השהייה במסך ניתן להעלים מתצוגת המסך את כפתורי העריכה \ הוספה \ מחיקה וזאת במטרה להקל על המשתמש, מבחינת כמות הפקדים במסך, או כדי למנוע טעויות שהמשתמש יכול לבצע במערכת.

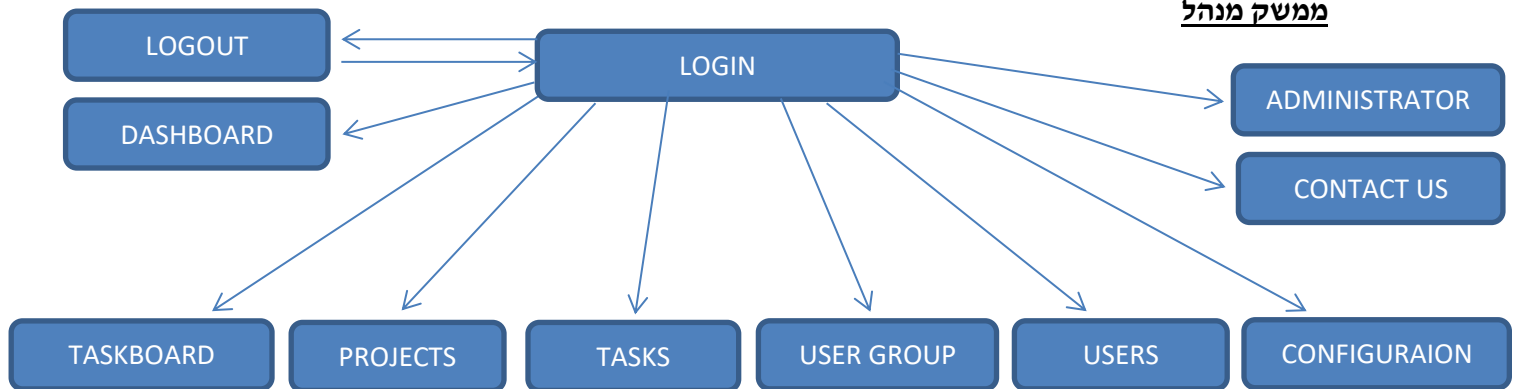
קיימת אחידות ועקביות בין המסכים השונים במערכת, תפעול תפריט מערכת הינו פשוט ומובן, הכנסת הנתונים לבית הנתונים פשוטה וברורה, זמני התגובה במערכת מהירים מאוד.

לאורך כל השהייה במערכת קיים סרגל כלים בחלק העליון של כל מסך שהמשתמש יבחר לגשת אליו, מאותו סרגל כלים, ניתן לדלג למסכים אחרים, לפנות לאדמיניסטרציה, או יצירת קשר.

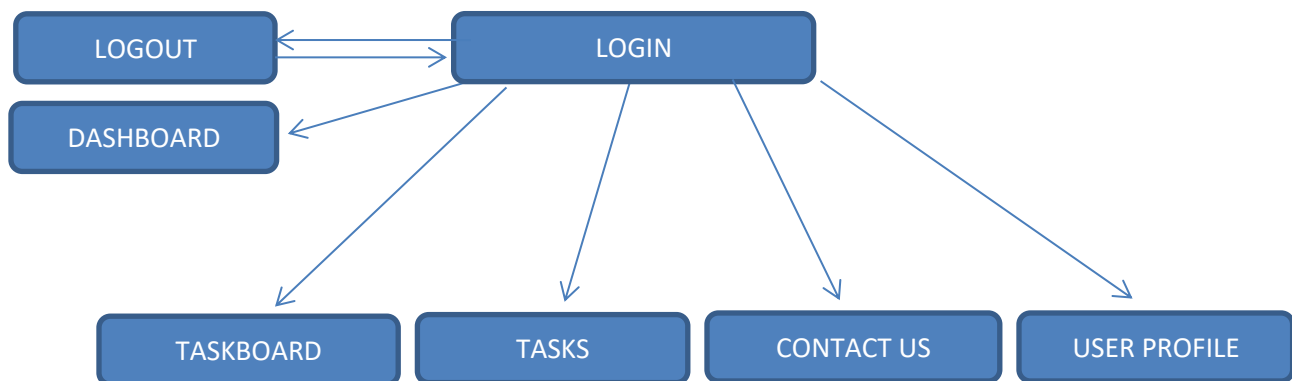
בנינו ממשק מובן, ידידותי, וקל לתפעול על ידי המשתמש, כל האינפורמציה הרלוונטית עבור כל מסך מוצגת באופן ברור.

**3.2 תיאור מסכים**

(לכל מסך תיאור משלו)

**ממשק מנהל**

משתמש מנהל יכול לגשת לכל מסך במערכת: ניהול פרויקטים, ניהול משמשים, ניהול משימות, ניהול קבוצות משתמשים, ניהול המערכת, וכמובן להוסיף \ למחוק \ לעדכן בכל מסך. בהמשך נפרט עבור כל מסך, מה מטרתו ומה האפשרויות לביצוע.

**ממשק משתמש רגיל**

למשתמש רגיל קיימות הרשאות לצפות בלוח המשימות, ולוח הפרויקטים, לערוך את פרטי הפרופיל שלו, ולגשת למסך צור קשר.

### 3.3 תיאור מסכים והסבר

#### Login - מסך כניסה למערכת

1. הרשמה עבור משתמשים חדשים
2. התחברות עבור משתמש קיים

#### Menu - תפריט

תפריט קבוע בחלקו העליון של המסך, שקיים בכל המסכים, ודרכו ניתן לנווט בין המסכים במערכת.

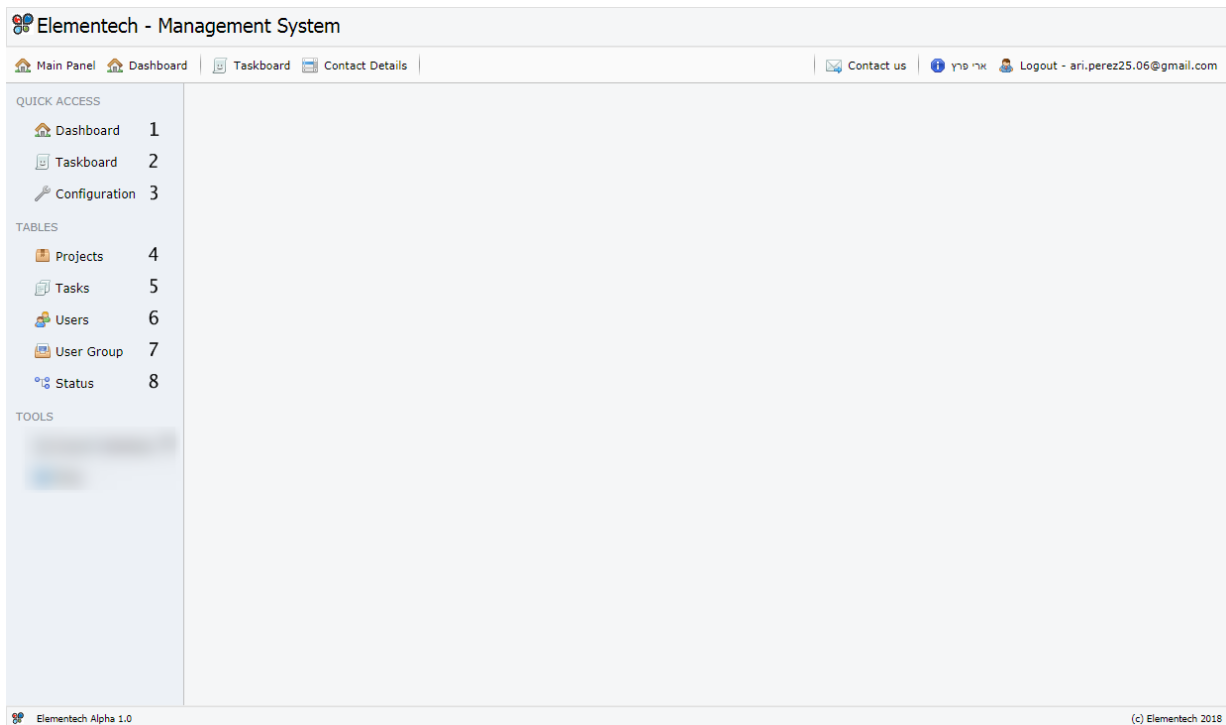
0. Main Panel – פקד שמפנה לתפריט רק עבור אדמין / מנהל / משתמש עם הרשאה בצפיית טבלאות.
1. Dashboard - צפייה בכלל הפעילויות במערכת.
2. Taskboard - צפייה בלוח משימות אישיות.
3. Contact Details – יומן אנשי קשר של כל העובדים.
4. Contact Us - צור קשר.
5. Profile edit – עריכת פרופיל משתמש.
6. בעת לחיצה על פקד זה, תצא מהמערכת ומסך ה- Login יוצג.



בהמשך נפרט עבור כלל המסכים.

## Main Panel – תפריט עבור אדמין / מנהל

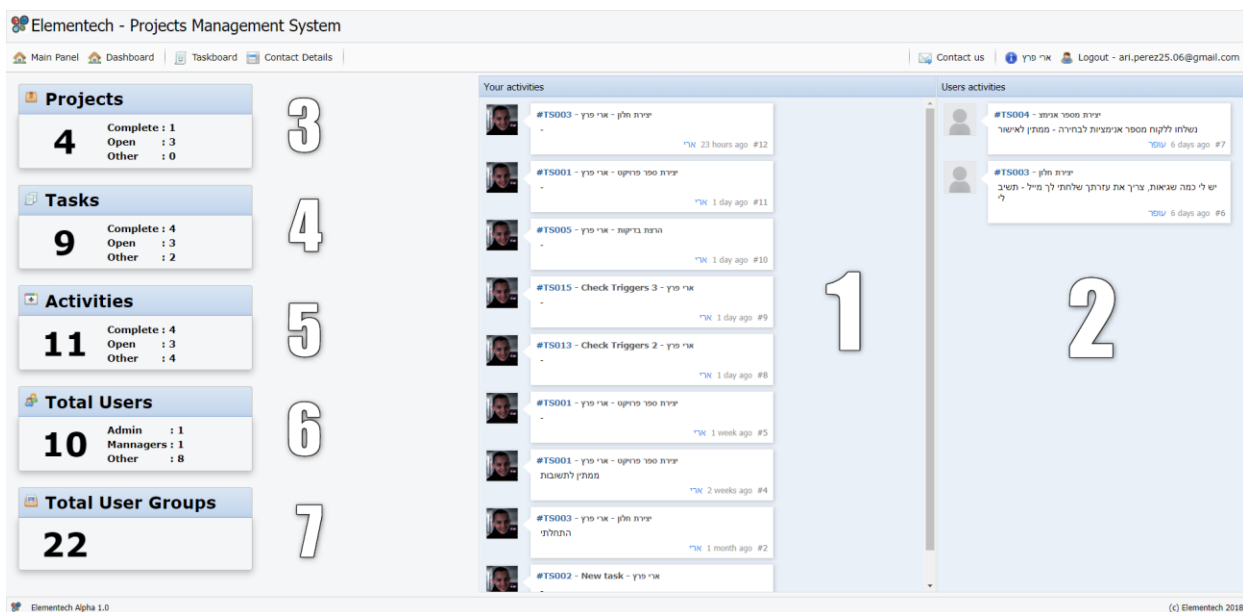
1. Dashboard – צפייה בכלל הפעילויות במערכת.
2. Taskboard – צפייה בלוח משימות אישיות.
3. Configuration – צפייה בדף הגדרות (מוצג רק לאדמין).
4. Projects – צפייה בטבלת פרויקטים (מוצג למנהל רק אם אדמין מאשר זאת).
5. Tasks – צפייה בטבלת משימות (מוצג למנהל רק אם אדמין מאשר זאת).
6. Users – צפייה בטבלת משתמשים (מוצג למנהל רק אם אדמין מאשר זאת).
7. User Group – צפייה בטבלת מחלקות (מוצג למנהל רק אם אדמין מאשר זאת).
8. Status – צפייה בטבלת סטטוסים (מוצג למנהל רק אם אדמין מאשר זאת).



## Dashboard – לוח מחוונים

במסך זה ניתן לצפות בסטטוס פרויקטים, סטטוס משימות, סטטוס פעילויות, עדכוני משתמשים עבור פעילויות, וכפי שציינו מעלה, מכל מסך ניתן לגשת לכל מסך אחר כל ידי תפריט קבוע המופיע בחלק העליון של המסך.

1. עדכוני פעילויות של משתמש.
2. עדכוני פעילויות של משתמשים.
3. סכום כללי של פרויקטים.
4. סכום כללי של משימות.
5. סכום כללי של פעילויות.
6. סכום כללי של משתמשים (מוצג רק עבור אדמין).
7. סכום כללי של מחלקות (מוצג רק עבור אדמין).



## Taskboard – לוח משימות

במסך זה ניתן לצפות בסטטוס המשימות האישיות של משתמש, לשנות את הצגת המשימות במסך על ידי בחירת סטטוס מסוים עבור משימה.

1. פקד רשימה נפתחת – אפשרות זו נותנת להציג את כל המשימות עבור אותו משתמש, ואת כל המשימות שהושלמו.
2. חלק זה במסך מציג את המשימות שהמשתמש קיבל, כל משימה נמצאת תחת שם הפרויקט המתאים שלה.
3. שם הפרויקט.
4. תיאור הפרויקט.
5. סטטוס פרויקט.
6. משימות הקשורות לאותו פרויקט.

לחיצה על מספר המשימה (פקד 1) תוביל למסך הבא, שבו ניתן להוסיף עדכונים עבור המשימה כדי שמשמשים אחרים יוכלו לצפות, להוסיף התקדמות עבור המשימה.

**מסך הצגת משימה והוספת עדכונים למשימה**

1. שם המשימה וקוד המשימה.
2. שם הפרויקט וקוד הפרויקט.
3. תיאור התקדמות הפרויקט עם תאריך התחלה ותאריך יעד ושם מפיץ המשימה.
4. סטטוס המשימה.
5. רשימת משתמשים שהמשימה הוקצתה להם.
6. תיאור המשימה.
7. פעילויות משתמשים.
8. פקדים לעריכה / מחיקת פעילות.
9. פקד להוספת פעילות.

The screenshot displays the Elementech Project Management System interface. At the top, the navigation bar includes links for Main Panel, Dashboard, Taskboard, and Contact Details. The user is logged in as 'ari.perez25.06@gmail.com'. The main content area shows a project titled '#TS001 - יצירת ספר פרויקט' (1). Below the title, project details are listed: Project: #PR002 - המכללה לימנהל אשדוד (2), Priority: High, Progress: 100%, Start Date: 07 January 2019 11:25, End Date: 07 January 2019 23:00, and Assigned by: ארי פרוץ (3). A table with two columns, Members and Description (4), lists the project members: ארי פרוץ (5) and a description in Hebrew (6). Below the table, a progress bar shows 100% completion (7). The bottom section displays a list of tasks (8) with their status, assigned to, and completion progress (9). The tasks are: '111' (13 June 2019 02:05, Complete, 100%), '112' (03 June 2019 13:40, Reopen, 50%), and '113' (31 May 2019 01:51, Open, 0%).







## Users - ניהול משתמשים

Elementech - Projects Management System

Main Panel

Dashboard

Taskboard

Contact Details

Contact us

ארי פרץ

Logout - ari.perez25.06@gmail.com

Users

Q: All Fields

Search

Add

Edit

Delete

Information

#	ID	Fullname	Nickname	Email	Phone	Password	Status	Permission Tables	Position	
1	SR000	ארי פרץ	ארי	ari.perez25.06@gmail.com	0505858108	Ap250691	Active	Active	Admin	
2	SR014	שפר חנוניה	שפר	Ofer@gmail.com	0524532524	123456Ae	Active	Active	Manager	
3	SR015	סחר נגב	סחר	Sahar@gmail.com	0505858108	Ac123456	Active	Nonactive	Python Expert	
4	SR016	ניב זאבון	ניב	niv@gmail.com	0505858108	NIV12345	Nonactive	Nonactive	C# Expert	
5	SR019	יואב כהן	יואב	Yoav@gmail.com	0505858106	YO123456	Active	Nonactive	Backend Programmer	
6	SR020	יצחק מור	יצחק	itzhak@gmail.com	0505656107	IM123456	Active	Nonactive	IT CARE	
7	SR021	אורלי שוב	אורלי	Orly@gmail.com	0505858108	OS123456	Active	Nonactive	Java Expert	
8	SR022	אנטוני פיזו	אנטוני	Antony@gmail.com	0504848594	Ap123456	Active	Nonactive	Help Desk	
9	SR023	ירון לעידות	ירון	Yaron@gmail.com	0505858108	YL123456	Active	Nonactive	SUPPORT ENGINEER	
10	SR024	מורן פרץ	מורן	moran@gmail.com	0505454359	Mb123456	Active	Active	CPA	

## Configuration - קונפיגורציית האתר

1. מציין כי המערכת במצב תחזוקה כך שמשתמשים יהיו מודעים לכך אם במידה ויש קריסות בזמן העבודה.
2. הרשאת הרשמה עבור משתמשים חדשים (מוסיף פקד REGISTER במסך ה-LOGIN).
3. הרשאת צפייה בטבלת משתמשים.
4. הרשאת צפייה בטבלת קבוצות משתמשים.
5. הרשאת צפייה בטבלת פרויקטים.
6. הרשאת צפייה בטבלת משימות.
7. הרשאת צפייה בטבלת סטטוסים.
8. בחירת כיוון הדף (מימין לשמאל, משמאל לימין).
9. שם המערכת.
10. שם הכותרת התחתונה.
11. פורמט התאריך והשעה (צריך להיות בפורמט של PHP).

Site configuration

Maintenance mode *	<input type="text" value="No"/>	*For managers	1
Guest register *	<input type="text" value="No"/>	*Allow guests to sign up	2
Users Tables *	<input type="text" value="Yes"/>	*Give managers permission to view users tables	3
Group Tables *	<input type="text" value="Yes"/>	*Give managers permission to view user groups tables	4
Projects Tables *	<input type="text" value="No"/>	*Give managers permission to view projects tables	5
Tasks Tables *	<input type="text" value="No"/>	*Give managers permission to view tasks tables	6
Status Tables *	<input type="text" value="No"/>	*Give managers permission to view status tables	7
Direction Page *	<input type="text" value="Left to right"/>	*Here you choose the direction of your page.	8
Site name *	<input type="text" value="Elementech - Projects Management"/>		9
Additional footer	<input type="text" value="(c) Elementech 2018"/>	*footer text	10
Datetime format	<input type="text" value="d F Y H:i"/>	*PHP Format	11
<input type="button" value="Save"/>			12

## Contact Us - יצירת קשר

ניתן לכתוב כל הודעה עבור מנהלי המערכת, המסך מחולק ל-3 מסכי קטגוריות, כללי, כתובת, ההודעה עצמה.

1. יש למלא שם פרטי, שם משפחה (שדות חובה) והערות.
2. יש למלא את כתובת המשתמש (שדות חובה).
3. יש למלא את ההודעה עצמה.
4. ניתן ללחוץ על הפקד כדי לנקות את הערכים שהוזנו.
5. ניתן ללחוץ על הפקד כדי לשלוח את ההודעה.

**MAIL TO US**

Your email: ester@gmail.com

1

Contact us

General Address Message

Name: \*

Last name: \*

Notes:

4 5

Clean the form Submit

**MAIL TO US**

Your email: ester@gmail.com

2

Contact us

General Address Message

Address: \*

Another address:

City: \*

State: \*

Postal Code: \*

Clean the form Submit

**MAIL TO US**

Your email: ester@gmail.com

3

Contact us

General Address Message

Short biography:

סמלית ההודעה: \*

The message title:

Clean the form Submit

## Profile editor – מסך עריכת פרופיל המשתמש

מסך לעריכת פרטי המנהל, הוספת תמונה, צפייה בפרויקטים שמעורב המנהל.

עבור כל מנהל קיים מספר סידורי במערכת השמור בבסיס הנתונים, שם משתמש מלא, כינוי, כתובת, מייל המשמש לכניסה למערכת.

My profile

General

ID: SR000

Fullname: \*

ארי פרץ

Nickname: \*

ארי

Address:

Shevet Gad 1/4

Password: \*

Ap250691

More Details

Email: \*

ari.perez25.06@gmail.com

Phone:


0505858108

About me:


-


Status:


Active

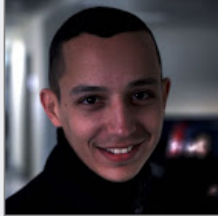
Position:  Admin

Member of projects

 המכללה למנהל אשדוד

 מפת חום

 פרויקט חלונות



Select a profile photo

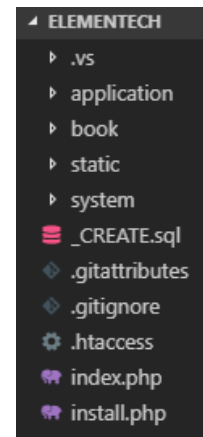
Save

## 4. מבנה נתונים וארגון הקבצים

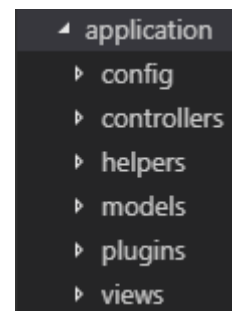
### 4.1 ארגון קבצים

#### 4.1.1 כללי

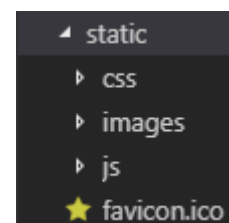
בתוך תיקיית הפרויקט קיים קובץ DB שבעצם מקים את בסיס הנתונים.



בתוך תיקיית ה-application קיימות תיקיות הפרויקט, views, models, controllers, אשר מכילות את מסכי המערכת, לוגיקה, פונקציות שונות במערכת.



בתוך תיקיית ה-static קיימות תיקיות העיצוב, המכילות את התמונות שיש במערכת, ותבניות העיצוב.



בסיס הנתונים מורכב מטבלאות שונות, והקשרים ביניהם מוצגים ב- ERD הבא.

The screenshot displays the phpMyAdmin interface for a database named 'elementech'. The left sidebar shows the database structure with a tree view. The main panel shows the 'Structure' tab, listing the tables in the database. Below the table list, there is a 'Create table' form with fields for 'Name' and 'Number of columns'.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> activity		11	InnoDB	latin1_swedish_ci	64 K	-
<input type="checkbox"/> config		1	InnoDB	latin1_swedish_ci	16 K	-
<input type="checkbox"/> project		4	InnoDB	latin1_swedish_ci	32 K	-
<input type="checkbox"/> runningnumber		4	InnoDB	latin1_swedish_ci	16 K	-
<input type="checkbox"/> status		9	InnoDB	latin1_swedish_ci	16 K	-
<input type="checkbox"/> task		9	InnoDB	latin1_swedish_ci	64 K	-
<input type="checkbox"/> tasker		13	InnoDB	latin1_swedish_ci	48 K	-
<input type="checkbox"/> user		10	InnoDB	latin1_swedish_ci	48 K	-
<input type="checkbox"/> usergroup		23	InnoDB	latin1_swedish_ci	16 K	-
<b>9 tables</b>	<b>Sum</b>	<b>84</b>	<b>InnoDB</b>	<b>latin1_swedish_ci</b>	<b>320 K</b>	<b>-</b>

Below the table list, there is a 'Create table' form with the following fields:

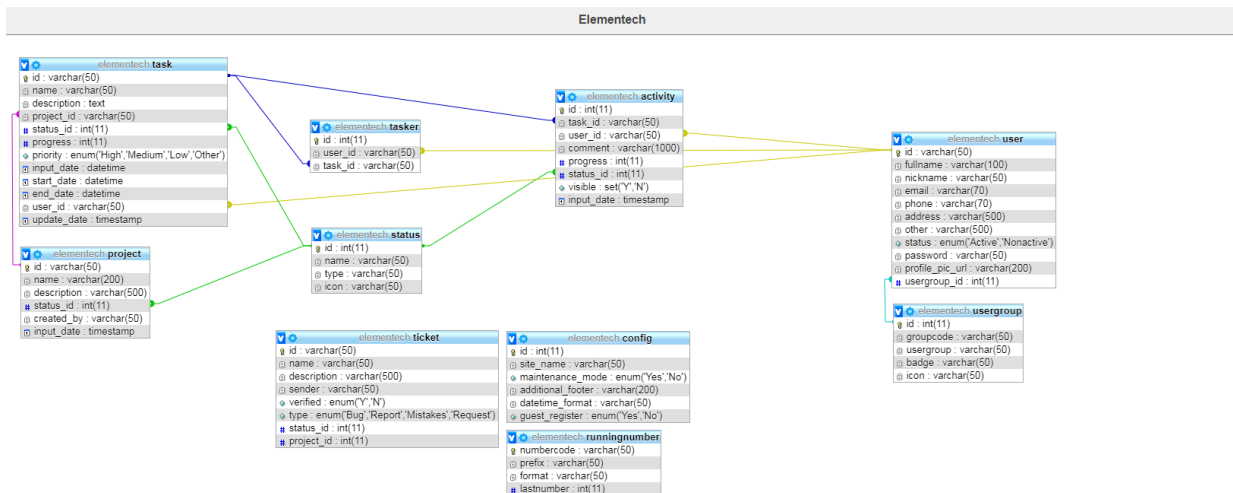
- Name:
- Number of columns:
- Go button



## 4.2 מבנה נתונים

תרשים ERD של בסיס הנתונים שמכיל נתונים אודות משתמשים, פרויקטים, משימות ועוד.

- **טבלת Task** – מספר משימה, שם משימה, תיאור, התקדמות, עדיפות (High, Medium, Low), תאריך הקלט, תאריך התחלה, תאריך סיום, תאריך עדכון, מספר משתמש (FK), מספר פרויקט (FK), סטטוס (FK).
- **טבלת Activity** – מספר פעילות, הערה, התקדמות, נראה (N\Y), תאריך הקלט, מספר משימה (FK), מספר משתמש (FK), סטטוס (FK).
- **טבלת Config** – מספר הקונפיגורציה, שם האתר, תחזוקה (N\Y), הוספת footer, פורמט התאריך, אורח (N\Y), הצג טבלת משתמשים (N\Y), הצג טבלת קבוצות משתמשים (N\Y), הצג טבלת פרויקטים (N\Y), הצג טבלת סטטוסים (N\Y), שנה כיוון מסכים (rtl\ltr).
- **טבלת Projects** – מספר פרויקט, שם פרויקט, תיאור, נוצר על ידי, תאריך הקלט, סטטוס (FK).
- **טבלת RunningNumber** – קוד מספר, קידומת, פורמט, מספר אחרון.
- **טבלת Status** – מספר סטטוס, שם סטטוס, סוג, אייקון.
- **טבלת Tasker** – מספר מבצע, מספר משתמש, מספר משימה.
- **טבלת User** – מספר משתמש, שם מלא, כינוי, מייל, טלפון, כתובת, אחר, סטטוס, סיסמא, הרשאת טבלאות, תמונה, משתמש בקבוצה (FK).
- **טבלת UserGroup** – מספר קבוצה, קוד קבוצה, שם משתמש, סמל, אייקון.
- בעזרת כלי Designer של MySQL ניתן לראות את הקשר שקיים בין הטבלאות.



## קטעי קוד - הגדרות מבני נתונים

```

1 CREATE TABLE `config` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `site_name` varchar(50) DEFAULT NULL,
4   `maintenance_mode` enum('Yes','No') DEFAULT NULL,
5   `additional_footer` varchar(200) DEFAULT NULL,
6   `datetime_format` varchar(50) DEFAULT NULL,
7   `guest_register` enum('Yes','No') DEFAULT NULL,
8   `SHOW_USERS_TABLES` enum('Yes','No') DEFAULT NULL,
9   `SHOW_USERSGROUPS_TABLES` enum('Yes','No') DEFAULT NULL,
10  `SHOW_PROJECTS_TABLES` enum('Yes','No') DEFAULT NULL,
11  `SHOW_TASKS_TABLES` enum('Yes','No') DEFAULT NULL,
12  `SHOW_STATUS_TABLES` enum('Yes','No') DEFAULT NULL,
13  `DIRECTION_PAGE` enum('ltr','rtl') DEFAULT NULL,
14  PRIMARY KEY (`id`)
15 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Config Table';
16
17 CREATE TABLE `usergroup` (
18   `id` int(11) NOT NULL AUTO_INCREMENT,
19   `groupcode` varchar(50) DEFAULT NULL,
20   `usergroup` varchar(50) DEFAULT NULL,
21   `badge` varchar(50) DEFAULT NULL,
22   `icon` varchar(50) DEFAULT NULL,
23   PRIMARY KEY (`id`)
24 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Group Table';
25
26 CREATE TABLE `runningnumber` (
27   `numbercode` varchar(50) NOT NULL,
28   `prefix` varchar(50) NOT NULL,
29   `format` varchar(50) NOT NULL,
30   `lastnumber` int(11) NOT NULL,
31   PRIMARY KEY (`numbercode`)
32 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
33
34 INSERT INTO `runningnumber` (`numbercode`, `prefix`, `format`, `lastnumber`) VALUES
35 ('PROJECT', 'PR', '000', 0),
36 ('TASK', 'TS', '000', 0),
37 ('USER', 'SR', '000', 1);
38
39 CREATE TABLE `status` (
40   `id` int(11) NOT NULL AUTO_INCREMENT,
41   `name` varchar(50) NOT NULL DEFAULT '0',
42   `type` varchar(50) DEFAULT NULL,
43   `icon` varchar(50) DEFAULT NULL,
44   PRIMARY KEY (`id`)
45 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

47 INSERT INTO `status` (`id`, `name`, `type`, `icon`) VALUES
48 (1, 'Open', 'PROJECT', 'icon-open'),
49 (2, 'Close', 'PROJECT', 'icon-close'),
50 (3, 'Reopen', 'PROJECT', 'icon-reopen'),
51 (4, 'Unknown', 'PROJECT', 'icon-unknown'),
52 (5, 'Complete', 'PROJECT', 'icon-complete'),
53 (6, 'Waiting Assesment', 'PROJECT', 'icon-waiting'),
54 (7, 'Stuck', 'PROJECT', 'icon-stuck'),
55 (8, 'Hold', 'PROJECT', 'icon-no');
56
57 CREATE TABLE `project` (
58   `id` varchar(50) NOT NULL,
59   `name` varchar(200) DEFAULT '-',
60   `description` varchar(500) DEFAULT '-',
61   `status_id` int(11) NOT NULL DEFAULT '0',
62   `created_by` varchar(50) NOT NULL,
63   `input_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
64   PRIMARY KEY (`id`),
65   KEY `FK_PROJECT_STATUS_ID` (`status_id`),
66   CONSTRAINT `FK_PROJECT_STATUS_ID` FOREIGN KEY (`status_id`) REFERENCES `status` (`id`)
67 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
68
69 CREATE TABLE `user` (
70   `id` varchar(50) NOT NULL,
71   `fullname` varchar(100) DEFAULT NULL,
72   `nickname` varchar(50) DEFAULT NULL,
73   `email` varchar(70) DEFAULT NULL,
74   `phone` varchar(70) DEFAULT NULL,
75   `address` varchar(500) DEFAULT NULL,
76   `other` varchar(500) DEFAULT NULL,
77   `status` enum('Active','Nonactive') DEFAULT NULL,
78   `PermissionTables` enum('Active','Nonactive') DEFAULT NULL,
79   `password` varchar(50) DEFAULT NULL,
80   `profile_pic_url` varchar(200) NOT NULL DEFAULT 'images/profile_pic_url/none.jpg',
81   `usergroup_id` int(11) DEFAULT NULL,
82   PRIMARY KEY (`id`),
83   UNIQUE KEY (`email`),
84   KEY `FK_G` (`usergroup_id`),
85   CONSTRAINT `FK_G` FOREIGN KEY (`usergroup_id`) REFERENCES `usergroup` (`id`) ON UPDATE CASCADE
86 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='User Table';
87
88 CREATE TABLE `task` (
89   `id` varchar(50) NOT NULL,
90   `name` varchar(50) NOT NULL DEFAULT '-',
91   `description` text,
92   `project_id` varchar(50) NOT NULL DEFAULT '0',
93   `status_id` int(11) NOT NULL DEFAULT '0',
94   `progress` int(11) NOT NULL DEFAULT '0',
95   `priority` enum('High','Medium','Low','Other') NOT NULL DEFAULT 'Medium',
96   `input_date` datetime DEFAULT NULL,
97   `start_date` datetime NOT NULL,
98   `end_date` datetime NOT NULL,
99   `user_id` varchar(50) NOT NULL,
100  `update_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP(),
101  PRIMARY KEY (`id`),
102  KEY `FK_TASK_STATUS_ID` (`status_id`),
103  KEY `FK_TASK_PROJECT_ID` (`project_id`),
104  KEY `FK_TASK_CREATED_BY` (`user_id`),
105  CONSTRAINT `FK_TASK_CREATED_BY` FOREIGN KEY (`user_id`) REFERENCES `user` (`id`),
106  CONSTRAINT `FK_TASK_PROJECT_ID` FOREIGN KEY (`project_id`) REFERENCES `project` (`id`) ON DELETE CASCADE ON UPDATE NO ACTION,
107  CONSTRAINT `FK_TASK_STATUS_ID` FOREIGN KEY (`status_id`) REFERENCES `status` (`id`)
108 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
109
110 CREATE TABLE `tasker` (
111   `id` int(11) NOT NULL AUTO_INCREMENT,
112   `user_id` varchar(50) NOT NULL DEFAULT '0',
113   `task_id` varchar(50) NOT NULL DEFAULT '0',
114   PRIMARY KEY (`id`),
115   KEY `FK_TASKER_TASK_ID` (`task_id`),
116   KEY `FK_TASKER_USER_ID` (`user_id`),
117   CONSTRAINT `FK_TASKER_TASK_ID` FOREIGN KEY (`task_id`) REFERENCES `task` (`id`) ON DELETE CASCADE ON UPDATE NO ACTION,
118   CONSTRAINT `FK_TASKER_USER_ID` FOREIGN KEY (`user_id`) REFERENCES `user` (`id`) ON DELETE CASCADE ON UPDATE NO ACTION
119 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

121 CREATE TABLE `activity` (
122   `id` int(11) NOT NULL AUTO_INCREMENT,
123   `task_id` varchar(50) NOT NULL DEFAULT '0',
124   `user_id` varchar(50) NOT NULL DEFAULT '0',
125   `comment` varchar(1000) DEFAULT '-',
126   `progress` int(11) DEFAULT NULL,
127   `status_id` int(11) DEFAULT NULL,
128   `visible` set('Y','N') NOT NULL DEFAULT 'Y',
129   `input_date` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
130   PRIMARY KEY (`id`),
131   KEY `FK_COMMENT_TASK_ID` (`task_id`),
132   KEY `FK_COMMENT_USER_ID` (`user_id`),
133   KEY `FK_COMMENT_STATUS_ID` (`status_id`),
134   CONSTRAINT `FK_COMMENT_TASK_ID` FOREIGN KEY (`task_id`) REFERENCES `task` (`id`) ON DELETE CASCADE ON UPDATE NO ACTION,
135   CONSTRAINT `FK_COMMENT_STATUS_ID` FOREIGN KEY (`status_id`) REFERENCES `status` (`id`),
136   CONSTRAINT `FK_COMMENT_USER_ID` FOREIGN KEY (`user_id`) REFERENCES `user` (`id`) ON DELETE CASCADE ON UPDATE NO ACTION
137 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
138
139 INSERT INTO `usergroup` (`id`, `groupcode`, `usergroup`, `badge`, `icon`) VALUES
140 (1, 'ADM', 'Admin', 'ADMINISTRATOR', 'icon-admin'),
141 (2, 'MAN', 'Manager', 'MANAGER', 'icon-manager'),
142 (3, 'PRO', 'Programmer', 'PROGRAMMER', 'icon-developer'),
143 (4, 'CPPEXP', 'CPPEXP', 'CPPEXP', 'icon-CPPEXP');

```

צילומי מסך של הטבלאות מתוך קטעי הקוד המתוארים מעלה

### Config table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 site_name	varchar(50)	latin1_swedish_ci		Yes	None			Change  Drop  More
<input type="checkbox"/>	3 maintenance_mode	enum('Yes', 'No')	latin1_swedish_ci		Yes	None			Change  Drop  More
<input type="checkbox"/>	4 additional_footer	varchar(200)	latin1_swedish_ci		Yes	None			Change  Drop  More
<input type="checkbox"/>	5 datetime_format	varchar(50)	latin1_swedish_ci		Yes	None			Change  Drop  More
<input type="checkbox"/>	6 guest_register	enum('Yes', 'No')	latin1_swedish_ci		Yes	None			Change  Drop  More
<input type="checkbox"/>	7 SHOW_USERS_TABLES	enum('Yes', 'No')	latin1_swedish_ci		Yes	None			Change  Drop  More
<input type="checkbox"/>	8 SHOW_USERSGROUPS_TABLES	enum('Yes', 'No')	latin1_swedish_ci		Yes	None			Change  Drop  More
<input type="checkbox"/>	9 SHOW_PROJECTS_TABLES	enum('Yes', 'No')	latin1_swedish_ci		Yes	None			Change  Drop  More
<input type="checkbox"/>	10 SHOW_TASKS_TABLES	enum('Yes', 'No')	latin1_swedish_ci		Yes	None			Change  Drop  More
<input type="checkbox"/>	11 SHOW_STATUS_TABLES	enum('Yes', 'No')	latin1_swedish_ci		Yes	None			Change  Drop  More
<input type="checkbox"/>	12 DIRECTION_PAGE	enum('ltr', 'rtl')	latin1_swedish_ci		Yes	None			Change  Drop  More

















### UserGroup Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 groupcode	varchar(50)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	3 usergroup	varchar(50)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	4 badge	varchar(50)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	5 icon	varchar(50)	latin1_swedish_ci		Yes	NULL			Change  Drop  More

**Status Table after insert**

<div><div><div></div><div></div><div></div></div></div>					id	name	type	icon		
<div><div><div></div></div></div>	<div><div><div></div></div></div>	Edit	<div><div><div></div></div></div>	Copy	<div><div><div></div></div></div>	Delete	1	Open	PROJECT	icon-open
<div><div><div></div></div></div>	<div><div><div></div></div></div>	Edit	<div><div><div></div></div></div>	Copy	<div><div><div></div></div></div>	Delete	2	Close	PROJECT	icon-close
<div><div><div></div></div></div>	<div><div><div></div></div></div>	Edit	<div><div><div></div></div></div>	Copy	<div><div><div></div></div></div>	Delete	3	Reopen	PROJECT	icon-reopen
<div><div><div></div></div></div>	<div><div><div></div></div></div>	Edit	<div><div><div></div></div></div>	Copy	<div><div><div></div></div></div>	Delete	4	Unknown	PROJECT	icon-unknown
<div><div><div></div></div></div>	<div><div><div></div></div></div>	Edit	<div><div><div></div></div></div>	Copy	<div><div><div></div></div></div>	Delete	5	Complete	PROJECT	icon-complete
<div><div><div></div></div></div>	<div><div><div></div></div></div>	Edit	<div><div><div></div></div></div>	Copy	<div><div><div></div></div></div>	Delete	6	Waiting Assesment	PROJECT	icon-waiting
<div><div><div></div></div></div>	<div><div><div></div></div></div>	Edit	<div><div><div></div></div></div>	Copy	<div><div><div></div></div></div>	Delete	7	Stuck	PROJECT	icon-stuck

**User Group Table after insert**

					id	groupcode	usergroup	badge	icon		
<input type="checkbox"/>		Edit		Copy		Delete	1	ADM	Admin	ADMINISTRATOR	icon-admin
<input type="checkbox"/>		Edit		Copy		Delete	2	MAN	Manager	MANAGER	icon-manager
<input type="checkbox"/>		Edit		Copy		Delete	3	PRO	Programmer	PROGRAMMER	icon-developer
<input type="checkbox"/>		Edit		Copy		Delete	4	CPPEXP	CPPEXP	CPPEXP	icon-CPPExpert
<input type="checkbox"/>		Edit		Copy		Delete	5	SCE	×j×•×>×Ÿ ×@×~×— SCE	SCE	icon-Python

## 5. תכנון

### 5.1 כללי ועקרונות התכנות (5.2)

Elementech היא מערכת ניהול פרויקטים מבוססת אינטרנט.

המערכת נכתבה בשפת PHP עם שימוש בטכנולוגיית ajax. השתמשנו בבסיס נתונים MySQL ולצורך עבודה עם בסיס הנתונים השתמשנו ב-JQuery, שהיא תוכנת קוד פתוח שאפשרה לנו ליצור אנימציות ואפקטים, לטפל באירועים במהלך שימוש המשתמש. בנוסף תוכנה זו משתמשת לפיתוח יישומי Ajax. כדי ליצור ממשק משתמש נוח, פשוט וידידותי, השתמשנו ב-W2UI שהוא ממשק משתמש מבוסס Java Script, תוך שימוש בטכנולוגיית CSS לעיצוב הממשק בצורה יפה, פשוטה ונוחה למשתמש.

בעת חשיבה על המערכת, רצינו שהמערכת תהיה פשוטה לתפעול, ושתציג את כל המידע שרלוונטי למשתמש במסך אחד, לצורך העניין, מנהל שמעוניין לצפות בפרויקטים, התקדמות, עדכוני משתמש, משימות פתוחות, משימות שהושלמו, יכול לצפות במסך Dashboard. כמו כן הייתה חשיבה על התחברות המשתמשים, לכן בחרנו במערכת מבוססת אינטרנט, חדשנית ונוחה שאינה מצריכה התקנה של תוכנת כבדות על המחשב המקומי.

את הפיתוח בצד השרת במערכת Elementech בחרנו לממש בעזרת תבנית עיצוב MVC (model-view-controller), התבנית מאפשרת חלוקה בין שלושה חלקים, מודל, תצוגה ובקר (ראה תרשים בסעיף 1.5), כך שכל מחלקה מבצעת את תפקידה, ושולחת מסרים למחלקה האחרת.

Model - במחלקה זו קיימות כל הפונקציות השולפות נתונים מה-Database, כותב וקורא מבסיס הנתונים, מטפל בחישובים שונים, אבטחת המערכת, במחלקה זו נראה בעיקר פונקציות של כתיבה וקריאה מבסיס הנתונים.

להלן קטע קוד מתוך מחלקת Model המטפלת בין היתר במשתמש.

```
<?php

class UserModel extends Model { //מחלקה יורשת/

    public function getAllUsers()
    {
        $result = $this->notorm()->user();
        return $result;
    }

    public function getUser($id = null)
    {
        if(empty($id)) die('Missing parameter. ');

        $result = $this->notorm()->user[$id];
        return $result;
    }

    public function getUserLogin($username = null, $password = null)
    {
        if(empty($username)) die('Missing parameter. ');
```

View - במחלקה זו מטפלים בכל הנוגע לממשק המשתמש, נראות המסך, מיקום המשתנים.

Controller - זה שמתווך בין השניים, ומעבד ומגיב לאירועים שביצע המשתמש בשכבת ה-view ובמידת הצורך יכול לשנות מידע בתצוגה. וכך נוצר הקשר בינו לבין ה-model.

### 5.3 תיאור אלגוריתמים

בכל מסך ראשי, לצורך העניין כמו Taskboard קיימת פונקציית index() אשר מציגה למשתמש את כל הנתונים הרלוונטיים למסך זה, במסך Taskboard בעת כניסה למסך, הפונקציה מציגה את כלל הפרויקטים וכלל הנתונים עבור כל פרויקט.

#### להלן פירוט של הפונקציות המרכזיות בכל מחלקה:

##### פונקציות הוספה

- הוספת פרויקט addProject(array)
- שמירת פרויקט saveProject(array)
- הוספת קבוצת משתמשים addUserGroup(array)
- הוספת משתמש addUser(array)
- הוספת פעילות insert\_activity(task\_id)
- הוספת משימה go\_insert\_task(id)
- הוספת משתמש חדש go\_insert\_user()
- הרשמה למערכת register()

##### פונקציית עדכון \ מחיקה

- עדכון משתמש go\_edit\_user(id)
- מחיקת משתמש delete\_user(id)
- עדכון פרופיל go\_edit\_profile(id)
- העלאת תמונת פרופיל upload\_profile\_pic(user\_id)
- עדכון משתמשי קבוצה go\_edit\_user\_group(array)
- מחיקת משתמש קבוצה delete\_user\_group()
- מחיקת פעילות deleteActivity(id)
- מחיקת פרויקט delete\_project(id)
- עדכון פרויקט go\_edit\_project(id)
- עדכון אתר בדף קונפיגורציה go\_edit\_config()
- מחיקת מבצע משימה על פי משימה deleteTaskerByTask(id)

שונות

- התנתקות מהמערכת logout()
- חיבור למערכת login()
- הצגת פרופיל משתמש profile(id)
- הצגת רשימת קבוצות משתמשים get\_user\_group\_list(array)
- שמירת משתמש קבוצה saveUserGroup(id)
- הצגת דף שגיאה () error404
- הצגת הפעילות האחרונה על פי משימה getLastActivityByTask(task\_id)
- הצגת פעילות על פי משימה getActivityByTask(task\_id)
- הצגת פרויקטים על פי משתמש getAllProjectsByUser(user\_id, type)
- הצגת התקדמות פעילות get\_activity(activity\_id)
- הצגת משימה על פי פרויקט ומשתמש  
getTaskByProjectAndUser(project\_id,user\_id,type)



• קטעי קוד מתועדים

```
class Activity extends Controller {

    function insert_activity($task_id = null) //פונקציה להוספת התקדמות של משימה
    {
        $_ACT = $this->loadModel('ActivityModel');
        $array = array( //מערך נתונים
            "id" => '',
            "task_id" => $task_id,
            "comment" => $_POST['record']['message'],
            "visible" => 'Y',
            "progress" => $_POST['record']['progress'],
            "status_id" => $_POST['record']['status_id']['id'],
            "input_date" => date('Y-m-d H:i:s'),
            "user_id" => Session::r('USER_ID')
        );
        $_ACT->saveActivity($array); //שמירת ההתקדמות
        $_TASK = $this->loadModel('TaskModel'); //אתחול
        $arrayTaskHeader = array( //עדכון הנתונים של משימה
            "id" => $task_id,
```

```
function go_edit_config() //פונקציה עריכת אתר בדף קונפיגורציה
{

    $_CONFIG = Handler::$_CONFIG;

    $array = array(
        "site_name" => $_POST['record']['site_name'],
        "additional_footer" => $_POST['record']['additional_footer'],
        "maintenance_mode" => $_POST['record']['maintenance_mode']['id'],
        "datetime_format" => $_POST['record']['datetime_format'],
        "guest_register" => $_POST['record']['guest_register']['id']
    );

    $_CONFIG->saveConfig($array);

    die();
}
```

```
class Taskboard extends Controller {

    function index()
    {
        $template = $this->loadView('taskboard_view');

        $_PROJECT = $this->loadModel('ProjectModel');
        $_TASK = $this->loadModel('TaskModel');

        $_PROJECT_DATA = $_PROJECT->getAllProjectsByUser(Handler::$_LOGIN_USER_ID);

        $result = array(); //הגדרת מערך

        $result[0] = array( //מערך שמחזיר את כלל הפרויקטים
            'id' => 'ALL',
            'text' => 'All projects',
            'img' => 'icon-box',
            'selected' => true,
            'type' => 'allproject',
```

## 6. מה הקנה הפרויקט

- ידע ומיומנות בתכנות בשפות JavaScript, PHP, וטכנולוגיות עיצוב שונות, CSS.
- למידת עקרונות ממשק אדם מכונה Human-computer Interface.
- מיומנות ביצירת בסיס נתונים ב-phpMyAdmin וניהולו.
- מיומנות וידע ביצירת תרשימי UML שונים, ERD, Use Case, Sequence Diagram.
- ניתוח מערכת תוך הגדרה והבנה של דרישות המערכת.
- הכרה ולמידה של ארכיטקטורות עיצוב שונות.
- ביצוע בדיקות ולידציה למערכת מסמך STP.
- שיתוף פעולה ועבודה בצוות תוך שימוש בטכנולוגיית GitHub.
- חקירת האינטרנט, למידה עצמית של נושאים שאינם מוכרים.

## 7. ביבליוגרפיה

### 7.1 ספרות

לא בוצע שימוש בספרי עזר.

### 7.2 אתרי אינטרנט

\* **Stack Overflow** – אתר אינטרנט פרטי פופולרי שנוצר ב-2008, העוסק התחום שאלות ותשובות בתחומי המחשוב ופיתוח תוכנה וחומרה. כל גולש יכול להירשם לאתר ולשאול שאלות או לתת תשובות.

\* **המדריך הישראלי ל-PHP**

\* **UDEMI**

\* **3wSchools**

## 8. נספחים

### 8.1 הוראות התקנה

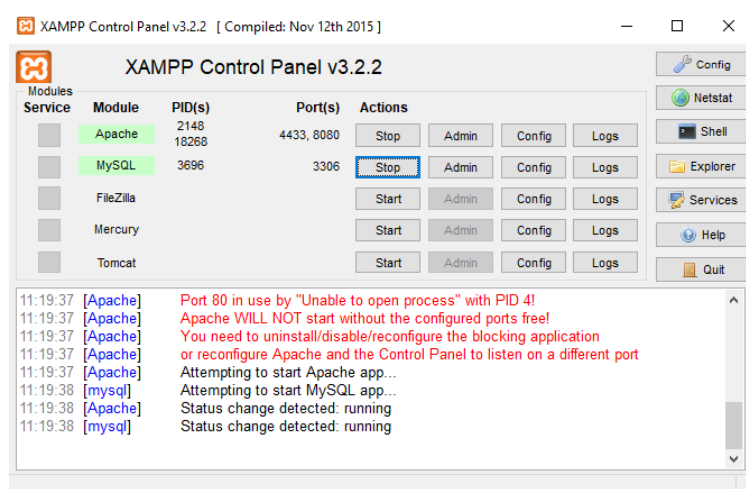
### 8.2 מדריך למשתמש

הוראות להתקנת המערכת :

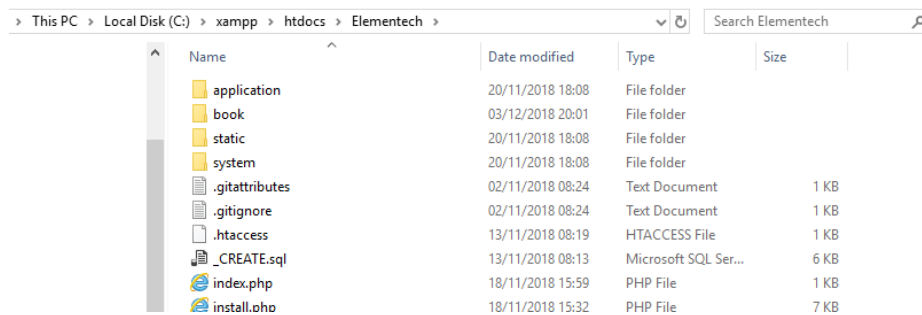
- יש להוריד ולהתקין את תוכנת XAMPP, ניתן להיעזר בלינק הבא :

<https://www.apachefriends.org/download.html>

- יש להדליק את ה-service, Apache ו-MYSQL.



- לאחר מכן יש לגרור את תיקיית הפרויקט Elementech לנתיב הבא : C:\xampp\htdocs :



- גש לנתיב: <http://localhost/elementech/install.php> ועקוב אחר ההוראות. מלא את כל השדות הדרושים.

**התקנת המערכת**

**כללי**

כתובת בסיס :  \*הסתבכת? דלג

שם האתר :  \*יופיע בכל ראש עמוד

שם בתחתית האתר :  \*יופיע בכל תחתית עמוד

פורמט התאריך :  \*פורמט תאריך זמן - מבוסס על PHP

**מסד נתונים**

מסד נתונים / מארח :  \*כתובת מסד הנתונים שלך

שם מסד נתונים :  \*שם מסד הנתונים שלך

שם משתמש :  \*שם המשתמש שלך

סיסמה :  \*הסיסמה שלך

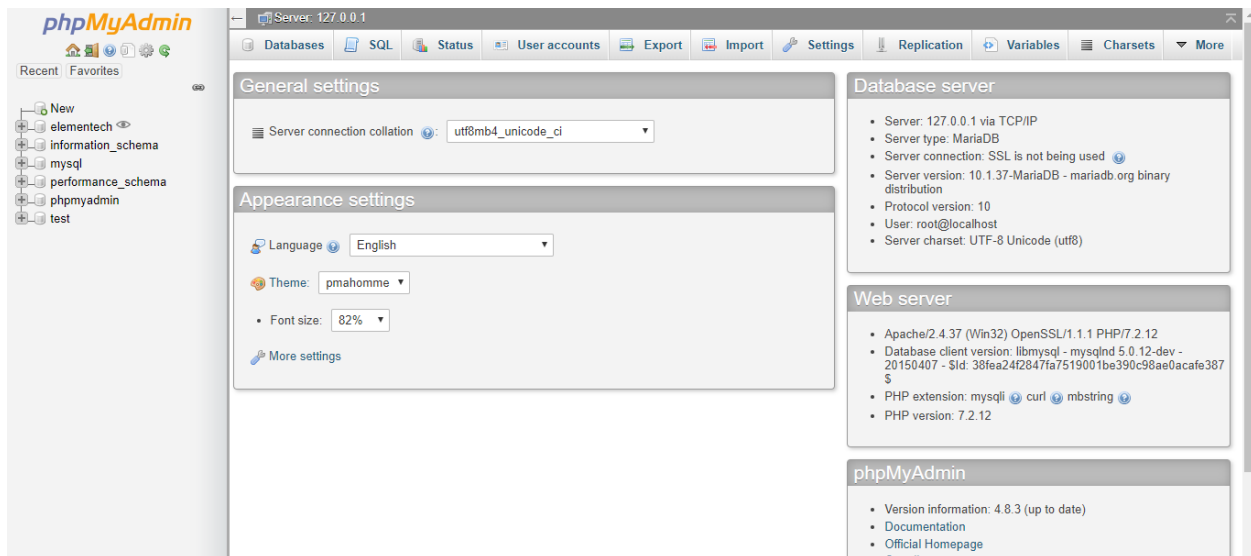
**ניהול האתר**

כתובת אימייל :

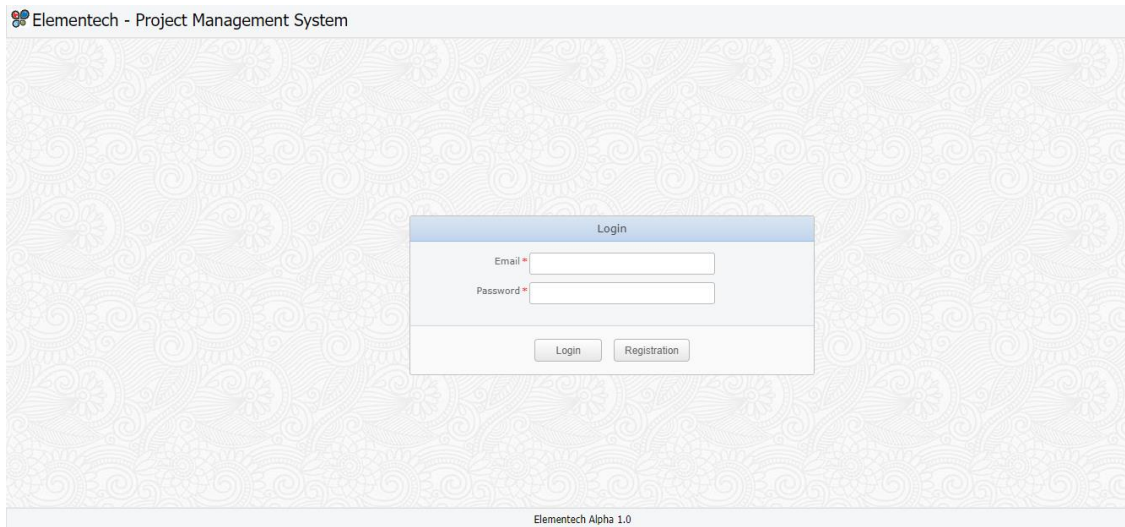
סיסמה :

[אשר נתונים](#)

- לאחר מכן ניתן להקליד בשורת ה-URL את הכתובת הבאה : [/http://localhost:8080/phpmyadmin](http://localhost:8080/phpmyadmin)
- כדי לצפות בבסיס הנתונים. בסיס הנתונים בשם Elementech מופיע בצד שמאל במסך.



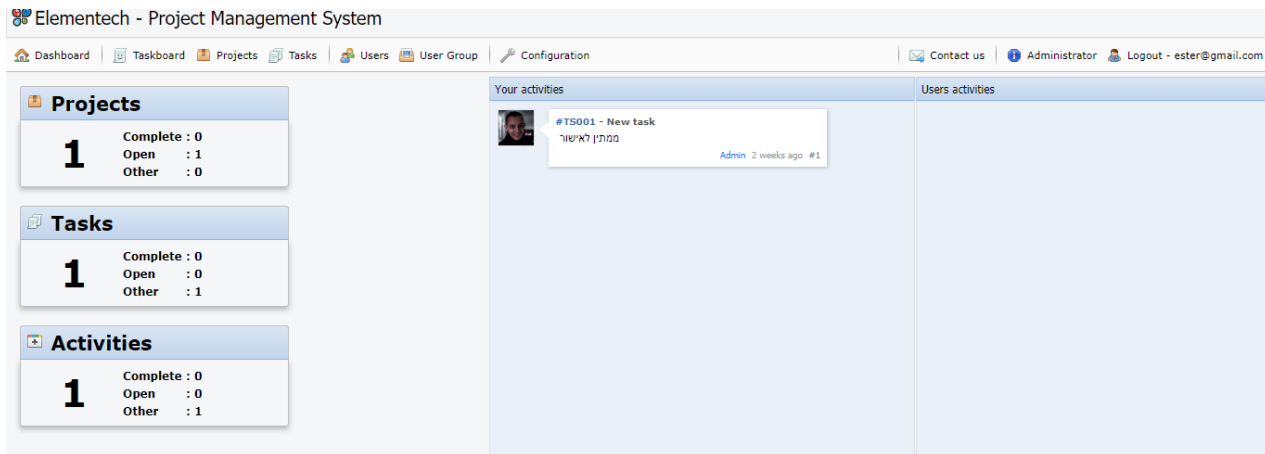
- כדי להיכנס למערכת ניהול הפרויקטים יש להכניס בשורת ה- URL את הכתובת הבאה :
- <http://localhost/MWS/Elementech/>



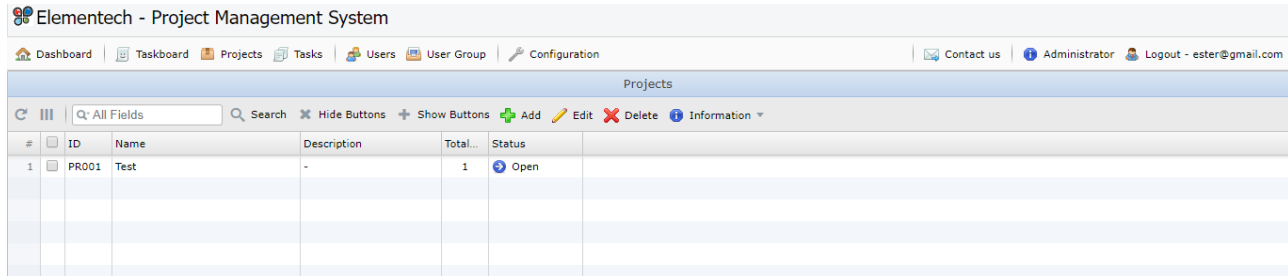
אימייל : [ari.perez25.06@gmail.com](mailto:ari.perez25.06@gmail.com) – או מה שקבע המנהל בהתקנת המערכת.

סיסמא : 250691 – או מה שקבע המנהל בהתקנת המערכת.

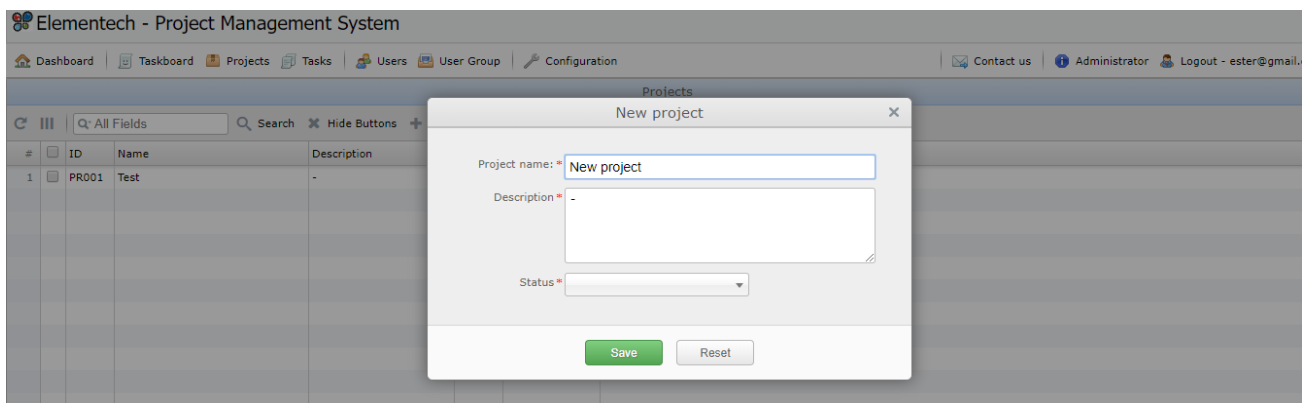
- לאחר מכן תועבר לדף אינדקס הבא :



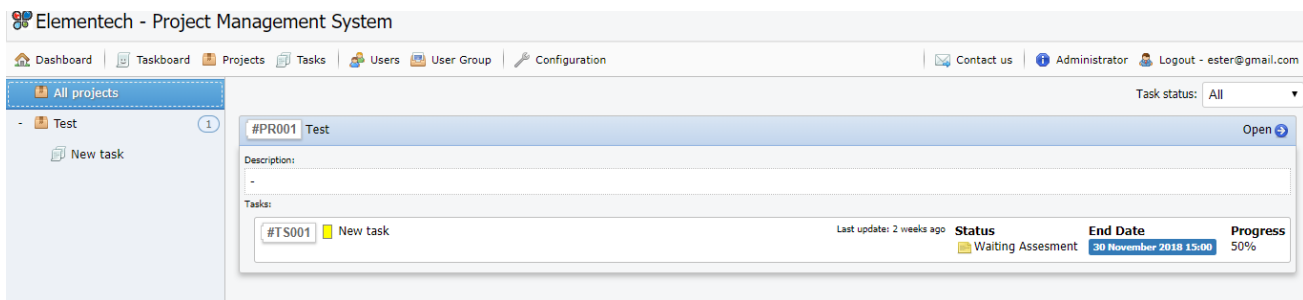
- כדי ליצור פרויקט חדש יש לגשת למסך Projects וללחוץ על המקש Add המופיע בירוק על המסך.



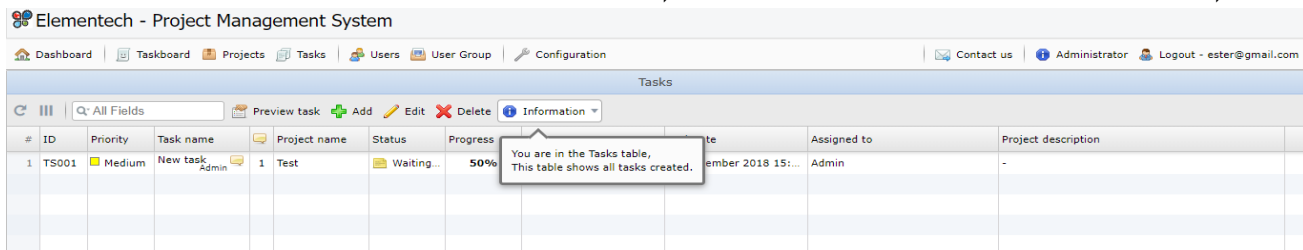
- לאחר לחיצה על כפתור ההוספה ייפתח חלון חדש שבו המשתמש יוכל להכניס פרטי פרויקט ולשמור.



- כדי לצפות בהתקדמות הפרויקט ניתן לגשת למסך Taskboard שמציג את כלל הפרויקטים, המשימות, התקדמות ונתונים נוספים אודות הפרויקט.



בכל מסך ישנו כפתור Information המציג מידע על המסך הנוכחי.



**8.3 קוד מקור מתועד****8.4 שונות**

NA

**פרקים מרכזיים נוספים לספר פרויקט:****9. דרישות מערכת ופונקציונליות.****9.1 דרישות לא פונקציונליות ודרישות מערכות.****הנחות ותלויות:**

המערכת תפעל במחשב בו קיימת מערכת הפעלה מגרסת windows 7 ומעלה ובתנאי שהתוכנה XAMPP שהיא תוכנה להתקנת שרת וירטואלי מותקנת בו לצורך הרצת ה-PHP ו-MYSQL.

**ממשק משתמש:**

- על המערכת לספק ממשק משתמש נוח, חדשני ופשוט לתפעול על ידי כל סוגי המשתמשים.
  - הממשקים יהיו בשפה האנגלית.
  - המערכת תנהל הרשאות, כך שלכל תפקיד בארגון תתאפשר גישה רק לפונקציונליות הרלוונטית לו. המערכת תחולק לשני ממשקים:
- א. **ממשק אדמין** - בעל יכולות עריכה וניהול משתמשים, ניהול פרויקטים, ניהול משימות ופעילויות.  
 ב. **ממשק משתמש רגיל** - משתמש בעל יכולות בסיסיות כגון צפייה בפרויקטים, צפייה במשימות.

**ממשק חומרה:**

- מערכת הפעלה WINDOWS 7 ומעלה.
- המערכת תתמוך בתוכנת XAMPP, ו-MyAdmin.php.

**ממשק תוכנה:**

- הטמעת המערכת בארגון, יש למדל את כלל התהליכים לתוך המערכת הממוחשבת בארגון. יש לוודא שהמערכת תהיה מוכנה לעבודה עם השינויים אותם נרצה להכניס וכי החומרה הקיימת מתאימה לעדכונים ותספק ביצועים ברמה גבוהה.
- המערכת מבוססת WEB הגישה תתאשר מכל מכשיר המחובר לרשת האינטרנט.
- המערכת תכתב בשפת תכנות PHP, HTML, תוך שימוש בטכנולוגיית CSS
- המערכת תעשה שימוש בספריית JS שנקראת W2ui שמטפלת ב-UI.
- המערכת תהיה רספונסיבית ומותאמת ל-Mobile על כל מאפייני התאימות בעזרת ספריית Bootstrap.
- המערכת מבססת מסד נתונים רציונלי MySQL.
- המערכת תתבסס על גרסת מערכת Windows 8.1 של חברת Microsoft ותתמוך בכל גרסאות מערכות ההפעלה שיותקנו במחשב לאחר מכן.

זמני תגובה

- על המערכת לפעול בזמני תגובה מהירים שלא יעלו על 2 שניות לפעולה.
- מהירות התחברות למערכת על ידי משתמש לא תעלה על 3 שניות.

אמינות וזמינות

- על המערכת לספק אבטחת המידע לפי התקן הקיים בשוק.
- המערכת תהיה זמינה באמצעות שרת פעיל 7\24 מכל מכשיר עם גישה לאינטרנט.
- יש לבצע גיבויים אוטומטיים למערכת על מנת לאפשר שיחזור של מידע במקרה של תקלה או מחיקת מידע בטעות.

גמישות

- המערכת תהיה גמישה לשינויים והוספת פיצ'רים בעתיד.

ביצועים

נדרוש שהמערכת תהיה יציבה וזמינה 24/7 למשתמשים, ללא קריסות או תקלות משביתות. התאוששות מקריסה תתבצע תוך זמן מהיר. מהירות תגובה במערכת בלחיצה על כפתור תהיה פחות מ 1 שניה לאחר הלחיצה.

אבטחה

שימוש במערכת יעשה רק בהתאם לסיסמאות ושם משתמש שניתנו לכל משתמש במערכת, נפרט מטה על חלופת אבטחת המידע שנבחר להשתמש.

9.2 דרישות פונקציונליות

- על המערכת לאפשר הוספת פרויקטים וקישור משימות בהתאם.
- על המערכת לאפשר למשתמש להתנתק בכל רגע נתון.
- על המערכת להציג למשתמש את המשימות שהוקצו לו.
- על המערכת לספק אופציה לפנות למנהלים בכל זמן שהוא.
- על המערכת לשמור היסטוריית פרויקטים ומשימות ופעילויות עבור משתמש.
- על המערכת לספק גישה למידע רלוונטי, dashboard פרויקטים ומשימות ומשתמשים.
- על המערכת לתת אפשרות של רישום משתמשים חדשים.
- על המערכת לתת למשתמש אפשרות להודיע על באג.
- על המערכת לתת למנהל אפשרויות ניהול משתמש ופרויקט ומשימות ופעילויות.
- על המערכת להציג סטטוס פרויקט.
- על המערכת להציג עבור כל משימה את ההתקדמות שלה.
- על המערכת לתת למשתמש לעדכן פעילויות ועדכונים במשימה.



## ממשק אדמין

### על ממשק האדמין חלות כלל הדרישות בממשק המשתמש ובנוסף:

- המערכת תאפשר הוספה/מחיקה של משתמשים.
- המערכת תאפשר לאדמין למחוק פרופיל משתמש.
- המערכת תאפשר הפסקת פעילות עבור פרויקט מסוים.
- המערכת תאפשר להוסיף/למחוק באופן ידני פרויקטים שהוכנסו לבסיס הנתונים.

## 9.3 חלופות ארכיטקטוניות

### 9.3.1 ברמת המערכת

#### א. ארכיטקטורה רב-שכבתית

ארכיטקטורת שרת-לקוח שבה ההצגה, העיבוד, וניהול הנתונים DB, הם תהליכים נפרדים מבחינה לוגית. השימוש הנפוץ ביותר בארכיטקטורה רב-שכבתית היא **ארכיטקטורת שלוש שכבות**.

מודל שלוש השכבות היא ארכיטקטורת שרת-לקוח שבה ממשק המשתמש, הלוגיקה, והגישה אל הנתונים מפותחים ומתוחזקים כמודולים נפרדים, בדרך כלל על גבי פלטפורמות שונות. מודל שלוש השכבות היא ארכיטקטורת תוכנה ותבנית עיצוב.

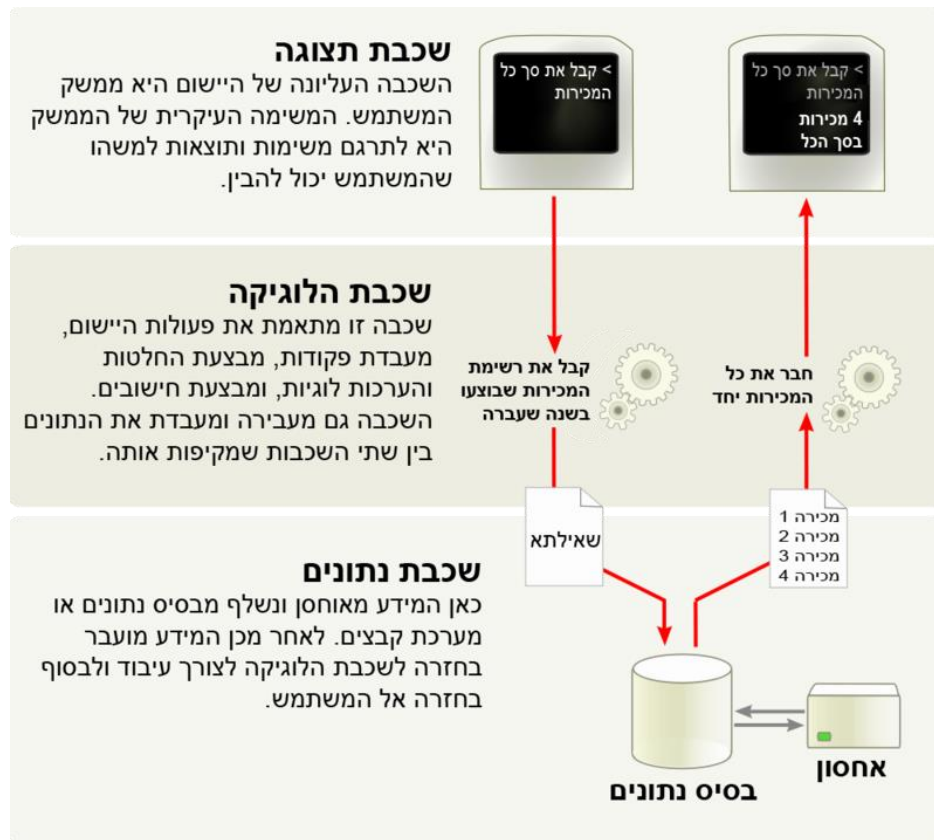
לארכיטקטורה זו ישנם יתרונות רבים ומשמעותיים, כמו תחזוקה, ניתן לבצע בדיקות בכל אחת מהשכבות בצורה נוחה ופשוטה, מקנה אפשרות להגדיר חוקים כללים עבור כל שכבה, ופשוטה ונוחה לעדכון הקוד, במיוחד שמדובר שעבודה של כמה מפתחים.

מלבד היתרונות הרגילים, ארכיטקטורת שלוש השכבות מאפשרת גם החלפה או שדרוג של כל אחת מהשכבות באופן בלתי תלוי בשכבות האחרות, בהתאם לדרישות או לשינויים בטכנולוגיה. לדוגמה, החלפת מערכת ההפעלה בשכבת התצוגה תשפיע רק על הקוד של ממשק המשתמש. בנוסף, ארכיטקטורה זו עוזרת למפתחים לשמור על פשטות לכל אורך פיתוח המערכת.

בשימוש בארכיטקטורה זו יש לשמור על הגיון בין השכבות, וכמובן לדעת לבודד כל שכבה, כדי "להנות" מהיתרונות שהצגנו למעלה.

לפי אתרי האינטרנט והמקורות השונים מומלץ להשתמש בארכיטקטורה זו כדי לבנות מערכות שצריכות להבנות מהירה, ובמיוחד מתאימה למפתחים חסרי ניסיון.

## ארכיטקטורת רב שכבתית - מודל שלוש השכבות



## ב. ארכיטקטורה מכוונת אירועים

ארכיטקטורה EDA, מוכוונת אירועים היא סגנון ארכיטקטוני המגדיר ארכיטקטורת תוכנה שהיחידות הבסיסיות שלה הן אירועים. השימוש בארכיטקטורה זו הוא במקרים רבים בשילוב עם ארכיטקטורה מוכוונת שירותים SOA.

כדי לממש ארכיטקטורה זו יש צורך בספק אירוע, ישות שמייצרת אירוע מסוים, ובצרכן אירוע, ישות המקבלת מסר על ההתרחשות ומידע מסוים, והיא בעצם מנתחת את האירוע ומגיבה עליו.

בהקשר של מערכות מחשוב המשמעות של אירוע היא מידע המועבר מרכיב תוכנה אחד לרכיב תוכנה אחר ודורש מהרכיב המקבל הפעלת תוכנית כלשהי.

קיים שימוש בארכיטקטורה זו המערכות של מסחר אלקטרוני, לצורך זיהוי ניסיון רמאות, לצורך העניין, בעת זיהוי האירוע, יש צורך לבצע פעולה אחרת בתגובה לאירוע שזוהה, ביטול עסקה, דווח לגורמי אכיפה, הכנסה לרשימה שחורה ועוד.

בכל מקרה ההגדרה של אירוע במונחים הללו היא אירוע עסקי, פונקציונליות עסקית, והצורך בקיום של אובייקט אירוע המחייב פעולה כלשהי, כמו העברת קובץ xml.

ישות צרכן האירוע חייב לדעת לזהות א האירוע ולהגיב בהתאם.

### **יתרונות בשימוש ארכיטקטורה זו:**

- ניתן להרחיב בקלות את המחלקות לאור הופעת אירועים חדשים.
- הסתגלות בקלות לסביבות מורכבות.

### **חסרונות בשימוש ארכיטקטורה זו:**

- קושי בביצוע בדיקות לאור העובדה שהמודולים תלויים אחד בשני.
- קושי בטיפול בשגיאות, כאשר השגיאה חוצה מודולים שונים במערכת.
- פיתוח מבנה נתונים מערכתית לאירועים עלול להיות מורכב יותר.

### **ארכיטקטורת מיקרו-שירותים Microservices architecture**

ארכיטקטורה זו היא שימושית מאד בשנים האחרונות, ומשמשת להמון מערכות web בתעשייה, אמזון, eBay נטפליקס ועוד המון ארגונים פופולריים משתמשים בה, וזה משום שהיא עוזרת ליצור חווית שימוש עבור המשתמש ומעניקה ממשק משתמש עשיר יותר.

בארכיטקטורה זו בעצם מפרקים את המערכת לרכיבים וכל רכיב עומד בפני עצמו וניתן לפתח אותו ללא תלות באחר, עבור כל רכיב מספיקים שני אנשי צוות שיעבדו ויתחזקו אותו, כל רכיב מבצע דבר אחד, ויש לו אחריות כל שירות אחד ויחיד מהמערכת כולה.

ארכיטקטורה זו משמשת לפיתוח מערכות מורכבות, בעלות בסיס נתונים רחב, ונותנת דגש משמעותי על חווית המשתמש. עוד יתרונות הם אמינות ותחזוקה, נפילה של רכיב אחד לא משביתה את כל המערכת, ביצוע שינויים במקביל על ידי מפתחים שונים, כמו כן לצד היתרונות ישנם חסרונות משמעותיים כמו, ביצוע רפלקציה במערכת כזו שהיא חוצה רכיבים, קריאות רבות של כל מיני רכיבים שעלולות להעמיס ולסבך את הקוד, קושי בשחזור בעיות וביצוע ניטור.

### **סיכום דיון חלופת ארכיטקטורות:**

במקרה שלנו, מערכת Elementech הינה מערכת עסקית שממשק המשתמש שלה הוא בעצם שיקוף של בסיס הנתונים, מערכת לניהול פרויקטים אשר מטרתה לעזור למנהלים ועובדים בארגון לנהל את הזמן שלהם, לעקוב אחר משימות, לייעל את התנהלות הארגון. ולצורך כך ארכיטקטורת שלוש השכבות הינה המתאימה ביותר למערכת שלנו.

### 9.3.2 ברמת מכלולי התוכנה:

**WordPress** - מערכת לניהול תוכן עם עיצובים מתקדמים המופצת באופן חופשי וחינמי. המערכת כתובה בשפת PHP על בסיס מסד נתונים MySQL. פלטפורמת WORDPRESS יודעת לעשות כמעט הכל, ובעלת הרחבות רבות. כוללת מאפיינים ותוספים רבים התואמים לכלל מערכות האתרים, ומאפיינים מתקדמים, התואמים למערכות ניהול תוכן מתקדמות. WORDPRESS עומדת בתקני רשת מקובלים, ידידותית למנועי חיפוש, מאפשרת מעקב אחרי שינויי עריכה, מכילה מערכת תגובות מובנית, מערכת לניהול קובצי מדיה ועוד. כמו כן ארכיטקטורת WORDPRESS תומכת בשפות רבות.

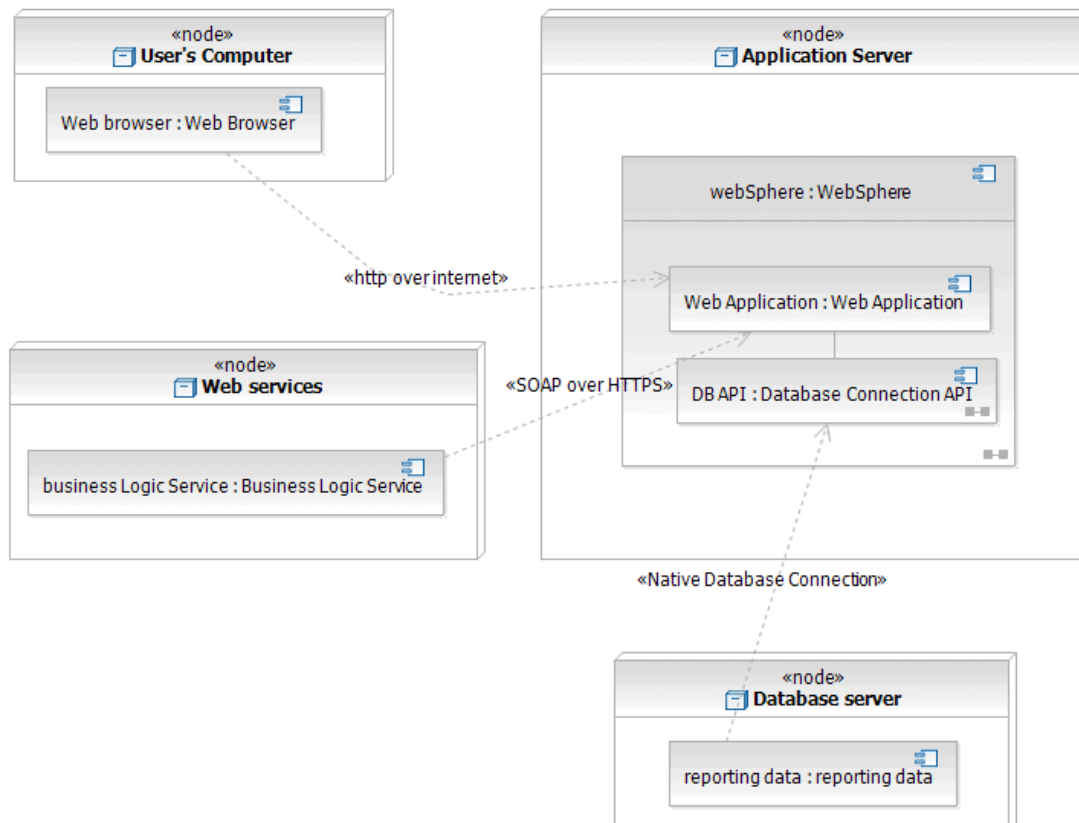
**WIX code** - מערכת שבאמצעותה ניתן לפתח בסיס נתונים דינמי, וכך להוסיף לאתר שבנית ב-WIX אין סוף תכנים שיוצגו לפי דרישת הגולש ולא צריך לבנות אותם עצמאית דף אחר דף. כלומר WIX פיתחו מערכת שמאפשרת שליפת נתונים מבסיס נתונים ובנייה שלהם בצורה אוטונומית בתוך דף אתר. האלמנט הזה חוסך זמן רב ומאפשר למשל לבנות המון מסכים. בנוסף, להבדיל מהעבר, קיימת אפשרות להתממשק עם מערכות חיצוניות, למלא טפסים ושאלונים בצורה קלה ופשוטה ללא תוספים מיוחדים.

שימוש בארכיטקטורת תוכנה זו מתאים בדרך כלל לחנויות אינטרנטיות שרוצות להציג קטלוג מוצרים, מקנות אפשרות לקנייה דרך האתר ועוד.

**במסגרת החשיבה על הצרכים ובניית המערכת שלנו**, ראינו לנכון כי אין צורך בארכיטקטורת תוכנה מיוחדת כמו שהצגנו לעיל, ניתן להשתמש ב-phpMyAdmin שזהו ממשק גרפי למשתמש (GUI) המשמש לניהול המידע בבסיס הנתונים MySQL. הממשק מבוסס web, נכנסים לבסיס הנתונים דרך הדפדפן, ממשק נוח קל וידידותי למשתמש, תומך בשפות רבות, ומאפשר לבצע מגוון רחב של פעולות כמו ייצוא של ERD, ניהול בסיס הנתונים, ייבוא קבצים ועוד. API פתוח, רישיון הינו חופשי וניתן לגשת לקוד המקור ולבצע שינויים בהתאמה. כמוכן שיש צורך בתמיכה של PHP בשרת.

## 10. טופולוגיית הפתרון הנבחר

בחרנו להציג את פרישת המערכת על ידי Deployment Diagram שהינו תרשים לייצוג רכיבים פיזיים במערכת, כך נוכל להציג את ההתממשקות ופרטי התוכנה בין הרכיבים. לעניות דעתנו הארכיטקטורה הינה מבוזרת, ז"א שקיים שרת מרכזי הנותן שירותים, וכל צומת היא בעלת מחשוב אוטונומי שנותנת שירותים מקומיים (קרי מחשבים).



## 11.1 ארכיטקטורה נבחרת

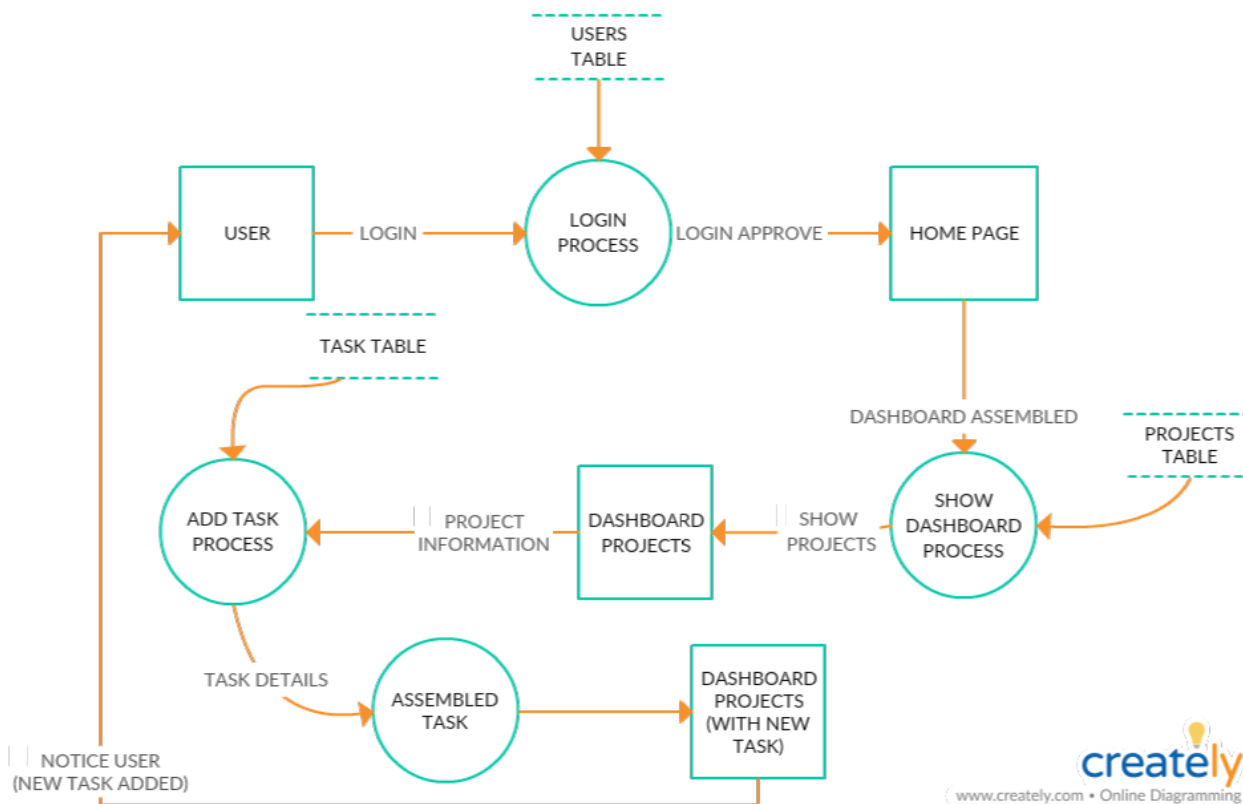
ארכיטקטורה EDA, מוכוונת אירועים היא סגנון ארכיטקטוני המגדיר ארכיטקטורת תוכנה שהיחידות הבסיסיות שלה הן אירועים. השימוש בארכיטקטורה זו הוא במקרים רבים בשילוב עם ארכיטקטורה מוכוונת שירותים SOA.

כדי לממש ארכיטקטורה זו יש צורך בספק אירוע, ישות שמייצרת אירוע מסוים, ובצרכן אירוע, ישות המקבלת מסר על ההתרחשות ומידע מסוים, והיא בעצם מנתחת את האירוע ומגיבה עליו.

### 11.1 ראה סעיף 1.5 תרשימי מערכת

#### Data Flow 11.1.1

הוספת משימה חדשה.



## 11.2 חלופות שפת מימוש

### חלופה 1:

#### PHP

שפה זו עומדת כמעט בכל הדרישות שהגדרנו לפני מימוש המערכת. שפה המותאמת לאתרי אינטרנט. שפה שמסוגלת להשתמש במשאבים של מחשב השרת, כמו שימוש בבסיס נתונים MySQL. שפה ליצירת דפי web דינאמיים וזאת כדי לנהל את ההרשמה \ כניסה \ נתונים שפה המסוגלת לנתח מידע ולתת פלט מתאים למשתמש. PHP הינה שפה ותיקה, קיימים באינטרנט המון מדריכים לשימוש בשפה זו. שפת open source. השפה הפופולרית השישית בעולם. פיתוח ישראלי, כך הנחנו שנמצא המון מדריכים בשפה העברית.

### חלופה 2:

#### Java Script

שפה זו הינה שפת סקריפט מבוססת אובייקטים, יש צורך לבצע שימוש ב-HTML ולהכניס קטעי קוד בתוך מסמכי HTML בכדי להפוך את הדף לדינמי, כמו שנדרש במערכת שלנו. נקודה זו נראתה לנו משמעותית מאד, והיותה בעצם הסיבה העיקרית לאי שימוש בשפה זו כשפת מימוש המערכת. כמו כן, כדי לדעת לתכנת בשפה זו יש צורך להכיר HTML ברמה גבוהה, דבר שלא הכרנו ברמה גבוהה.

דרישה נוספת שהייתה לנו טרם כתיבת המערכת היא שפה תואמת בצד השרת וצד הלקוח וכדי לפתח צד שרת עם שפה זו יש צורך בשימוש בטכנולוגיות נוספות כמו Node.js, ובנוסף לאחר חקירת השפה גילינו כי פיתוח java script בצד שרת מתחיל ממש בשנים האחרונות ולכן חששנו שלא נמצא מדריכים והסברים לשימוש בשפה רחבי באינטרנט.

לאור ניתוח הדרישות והמענה של שתי החלופות הנ"ל בחרנו בחלופה 1 שהיא מימוש בשפת PHP. ישנו שימוש מועט ב-javascript כדי שנוכל להשתמש בממשק המשתמש שרצינו UI2W, עיקר המערכת כתובה ב-PHP.

## **12. חלופות אבטחתיות והגנה**

בשוק קיימות המון תוכנות אנטי וירוס המציעות הגנה מפני סיכונים ואיומים, במיוחד כאשר ישנו שימוש באינטרנט.

### **אנטי וירוס ESAT**

תוכנה זו מציעה מגוון רחב של פתרונות הגנה מאיומים שונים, אפילו כאלה שעדיין לא הזיקו למחשב כמו, הגנה מפני חדירת קוד למערכת ההפעלה, מאגר מידע ענן שמתעדכן כל רגע ובודק קבצים במחשב, הגנה על החומרה במחשב, הגנה מפני התקפות ברשת, בדיקת כל אובייקט במחשב, הגנה מפני וירוס כופר, חוסם פרצות אבטחה ועוד. הדבר החשוב ביותר הוא שתוכנה זו משלבת בתוכה גם רכיב חומת אש (firewall) אשר מספק הגנה נוספת וחסימת התקשרויות לא רצויות. תוכנה זו אינה חינמית, והשימוש בה לאורך זמן דיי יקר.

### **אנטי וירוס AVG 8.5 Free**

אנטי וירוס בחינם המאפשר לעצור וירוסים, תוכנות ריגול ויישומים זדוניים רבים אחרים. התוכנה כוללת עדכוני אבטחה בזמן אמת, סריקות לגילוי איומים ותקלות בביצועים, וגם עוצר הורדות מסוכנות לפני שהם מגיעות למחשב. בנוסף התוכנה מציעה גלישה בטוחה ואבטחה על קבצי המייל. ההתקנה היא פשוטה, והשימוש אינו משפיע על ביצועי המחשב. תוכנה זו הינה חינמית.

### **חלופה נוספת היא שילוב של תוכנת אנטי וירוס חינמית הכוללת חומת-אש והתקנות אבטחה כאלה ואחרות.**

קודם כל לצורך אבטחה והגנה על בסיס הנתונים, יש להפריד בין שרת האינטרנט לבין שרת מסד הנתונים. כמו כן, החלופה תכלול שימוש באישורי אבטחה SSL, כדי לאבטח את דף הכניסה של phpMyAdmin על ידי מניעת העברת שם משתמש וסיסמא כטקסט רגיל ושמירה על הנתונים. בנוסף, phpMyAdmin מאפשר שימוש בכלי htpasswd כדי ליצור סיסמא עבור החשבון של המשתמש שהזין אותה וכך לשמור על הגנת הנתונים. כמו כן, ניתן להשבית את הגישה לחשבון המקורי (root account) במערכת phpMyAdmin.

## **12.1 פירוט בדירות תוכנה ואופן ביצוען DOC-STP**

### **תכנון בדיקות מערכת**

בדיקות המערכת יעשו בשיטת פיתוח מונחה בדיקות (TDD). תהליך זה פשוט ויעיל לפיתוח קוד באיכות גבוהה, ובכך יש למפתחים את היכולת לפתח במהירות וביעילות, תוך צמצום משמעותי של כמות השגיאות על כן, ייכתבו טסטים עבור כל מסך במערכת.

### **תכנון בדיקות יחידה**

בדיקות היחידה יתבצעו לאחר כל סיום מסך או לאחר כל שינוי שיתבצע במערכת. נבצע את הבדיקות באמצעות PHPUnit, כלי פתוח וחופשי לבדיקות יחידה בשפת PHP. אם במידה ואחד מהצוות נתקל בשגיאה, עליו לתקן את השגיאה במייד, ואין עליו להמשיך את כתיבת הקוד של המסך הבא כל עוד יש בעיה. על חבר הצוות לוודא כי הבדיקה רצה וכי אין שגיאות.



מס'	הסבר על הפונקציה הנבדקת	תיאור הפונקציה הבודקת	פרמטרים נדרשים לבדיקה	תוצאה צפויה	סטטוס
1	פונקציה אשר בודקת האם המשתמש "פעיל". הפונקציה מקבלת את מזהה המשתמש ובודקת אם הוא פעיל- אם כן תחזיר true אחרת false	הפונקציה תקבל משתמש עם סטטוס פעיל	checkStatus(id)	true	נבדק
2		הפונקציה תקבל משתמש עם סטטוס לא פעיל	checkStatus(id)	true	נבדק
3	פונקציה אשר מקבלת אירוע, נתוני משתמש שמנסה להתחבר. הפונקציה תקבל מספר 1 במידה והתחברות הצליחה, בפונקציה תקבל 2 במידה שאין הרשאה למשתמש להתחבר למערכת, הפונקציה תקבל 3 במידה שהוכנסו נתונים לא נכונים	בדיקת משתמש עם שם משתמש וסיסמא נכונים, אך מייל תפוס. הפונקציה תחסום את כניסת המשתמש למערכת.	function (event, data)	false (2)	נבדק
4		בדיקת משתמש עם שם משתמש וסיסמא נכונים. הפונקציה תאשר את כניסת המשתמש למערכת.	function (event, data)	true (1)	נבדק
5	פונקציה שמעדכנת את התקדמות המשתמש במשימה מסוימת. הפונקציה תקבל משתמש ומשימה, ותבדוק האם המשתמש תקין, תבדוק מהי הפעילות האחרונה במשימה ע"מ לקדם אותה, ותעדכן את המשימה בהתאם. במידה ואחד התנאים לא תקין, תקבל הודעת שגיאה.	בדיקת משתמש ללא משימות.	go_edit_activity (id, task)	false	נבדק
		בדיקת משתמש עם משימות, כאשר הפעילויות מלאות (ז"א לא ניתן לקדם).	go_edit_activity (id, task)	false	נבדק
		בדיקת משתמש בעל משימות לא מלאות.	go_edit_activity (id, task)	true	נבדק
6	פונקציה אשר מקבלת מזהה משתמש לצורך הצגתו במסך, הפונקציה תבדוק האם הפרמטר שהתקבל אינו ריק, ותפעיל את פרטי המשתמש	בדיקה האם הפרמטר אינו ריק, הפונקציה תקבל פרמטר אינו ריק.	Preview(id)	true	נבדק
7		בדיקה האם הפרמטר אינו ריק ושליחתו לפונקציה שמציגה את פרטי המשתמש והיא נכשלת	Preview(id) loadView(id)	false	נבדק
8		בדיקה האם למשתמש ישנם נתונים להצגה	Preview(id) loadView(id)	true	נבדק
9		בדיקה של קבלת פרמטר אינו מסוג ID	Preview(id)	false	נבדק