תוכן עניינים

[1. הגדרת דרישות ותאור כללי 3](#_Toc10731712)

[1.1 תיאור כללי 3](#_Toc10731713)

[1.1.1 מטרות המערכת 3](#_Toc10731714)

[1.1.2 היקף הפרויקט 3](#_Toc10731715)

[1.1.3 מבנה וארגון 3](#_Toc10731716)

[1.1.4 משימות המערכת 3](#_Toc10731717)

[1.2 תיאור חומרת המערכת 4](#_Toc10731718)

[1.2.1 כללי 4](#_Toc10731719)

[1.2.2 מרכיבי המערכת 4](#_Toc10731720)

[1.3 תיאור תוכנת המערכת 4](#_Toc10731721)

[1.3.1 כללי 4](#_Toc10731722)

[1.3.2 תיאור ה-SYSTEM 4](#_Toc10731723)

[1.3.3 כלי התוכנה לפיתוח המערכת 5](#_Toc10731724)

[1.4 תיאור פונקציות המערכת 5](#_Toc10731725)

[1.5 זרימת המידע במערכת 6](#_Toc10731726)

[1.5.1 כללי 6](#_Toc10731727)

[1.5.2 תרשימים 6](#_Toc10731728)

[1.5.3 מילון מונחים 9](#_Toc10731729)

[phpMyAdmin 9](#_Toc10731730)

[2. ממשקים חיצוניים 10](#_Toc10731731)

[3. ממשק אדם מכונה 10](#_Toc10731732)

[3.1 כללי 10](#_Toc10731733)

[3.2 תיאור מסכים 11](#_Toc10731734)

[3.3 תיאור מסכים והסבר 12](#_Toc10731735)

[4. מבנה נתונים וארגון הקבצים 23](#_Toc10731736)

[4.1 ארגון קבצים 23](#_Toc10731737)

[4.1.1 כללי 23](#_Toc10731738)

[4.2 מבנה נתונים 25](#_Toc10731739)

[5. תכנון 30](#_Toc10731740)

[5.1 כללי ועקרונות התכנות (5.2) 30](#_Toc10731741)

[5.3 תיאור אלגוריתמים 31](#_Toc10731742)

[6. מה הקנה הפרויקט 34](#_Toc10731743)

[7. ביבליוגרפיה 34](#_Toc10731744)

[7.1 ספרות 34](#_Toc10731745)

[7.2 אתרי אינטרנט 34](#_Toc10731746)

[8. נספחים 35](#_Toc10731747)

[8.1 הוראות התקנה 35](#_Toc10731748)

[8.2 מדריך למשתמש 35](#_Toc10731749)

[8.3 קוד מקור מתועד 39](#_Toc10731750)

[8.4 שונות 39](#_Toc10731751)

[9. דרישות מערכת ופונקציונליות. 39](#_Toc10731752)

[9.1 דרישות לא פונקציונליות ודרישות מערכות. 39](#_Toc10731753)

[9.2 דרישות פונקציונליות 40](#_Toc10731754)

[9.3 חלופות ארכיטקטוניות 41](#_Toc10731755)

[9.3.1 ברמת המערכת 41](#_Toc10731756)

[9.3.2 ברמת מכלולי התוכנה: 44](#_Toc10731757)

[10. טופולגיית הפתרון הנבחר 45](#_Toc10731758)

[11. ארכיטקטורה נבחרת 46](#_Toc10731759)

[11.1 ראה סעיף 1.5 תרשימי מערכת 46](#_Toc10731760)

[11.1.1 Data Flow 46](#_Toc10731761)

[11.2 חלופות שפת מימוש 47](#_Toc10731762)

[12. חלופות אבטחתיות והגנה 48](#_Toc10731763)

[12.1 פירוט בדירות תוכנה ואופן ביצוען STP-DOC 48](#_Toc10731764)

# 1. הגדרת דרישות ותאור כללי

## 1.1 תיאור כללי

### 1.1.1 מטרות 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 מושתת 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- JavaScriptו-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 שליפה, 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:

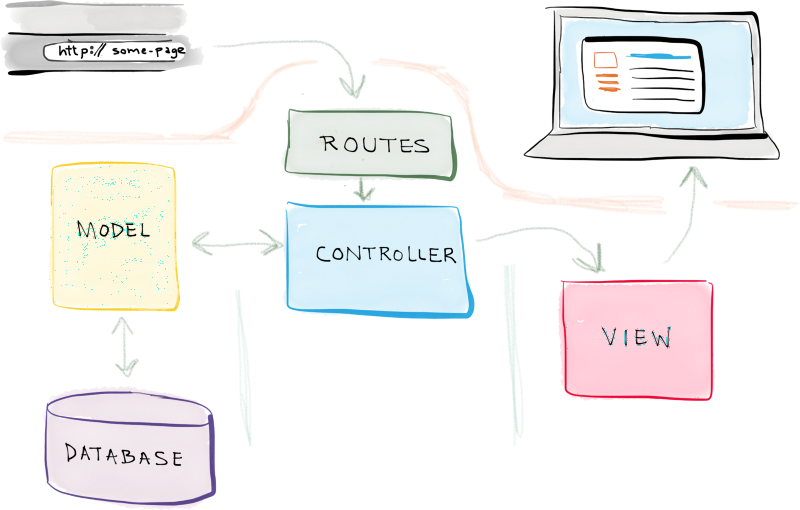
* חיפוש פרויקט \ משימה \ פעילות.
* עדכון נתוני פרויקט \ שם משימה.
* חיפוש פעילות על פי משימה.
* מחיקת פרויקט \ משימה \ פעילות.
* חיפוש משימה על פי פרויקט ומשתמש.
* שמירת פרויקט \ משימה \ פעילות.
* הוספת התקדמות משתמש.
* הצגת התקדמות משתמש.
* עריכת נתוני פרויקט \ משימה \ פעילות.

## 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.

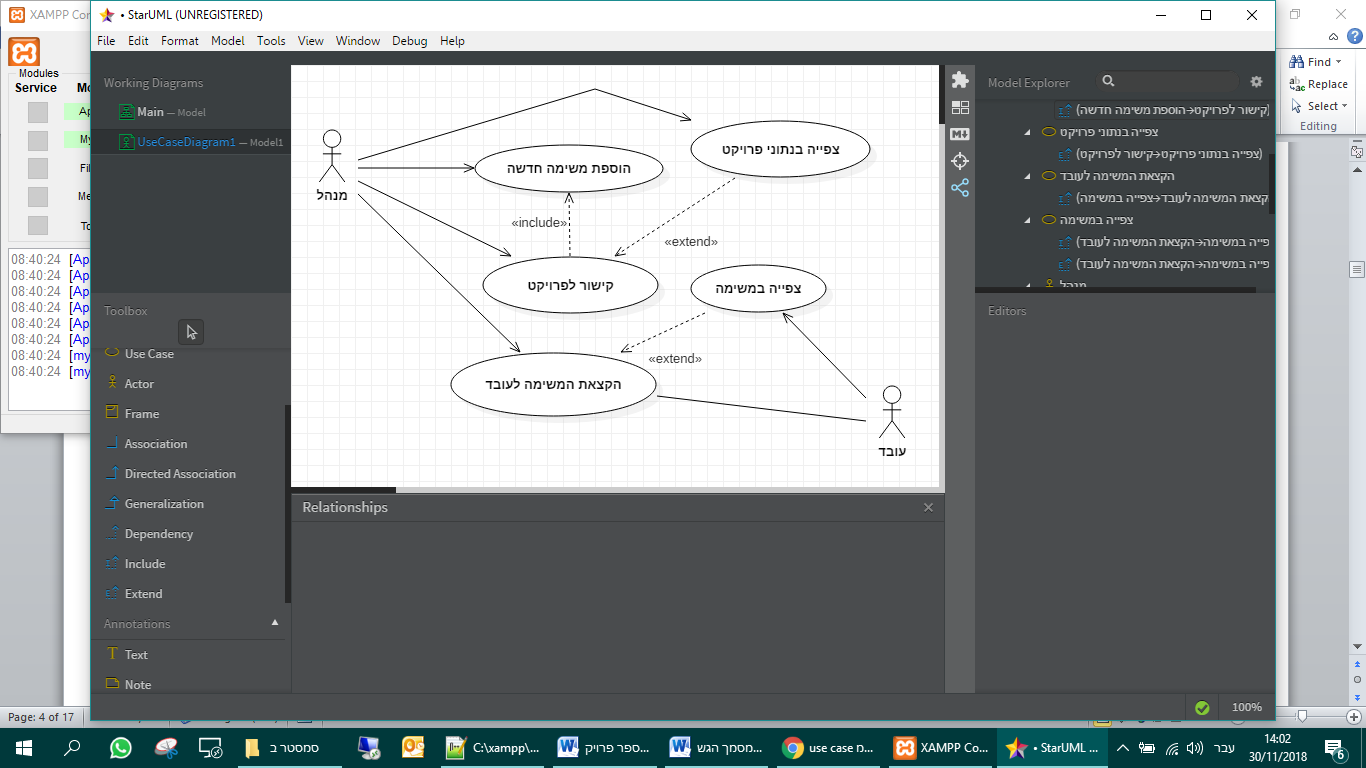
### 1.5.2 תרשימים



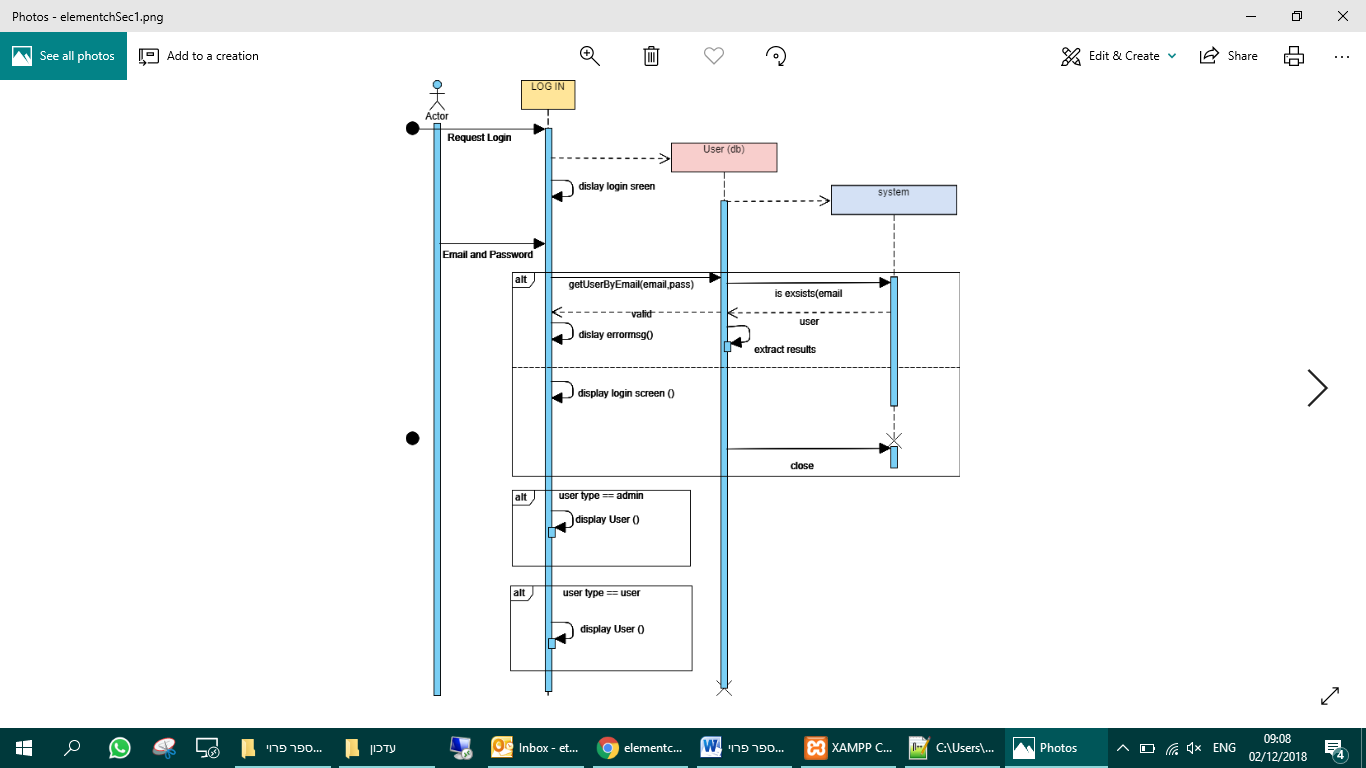
- Modelמטפל בקריאה ובכתיבת המידע למסד הנתונים ובכל הפעולות שקשורות לליבה העסקית של האפליקציה: חישובים למיניהם, אבטחה וסיסמאות וכל מה שהוא לא תצוגה.  
  
 - Viewכל מה שקשור לתצוגה, ה-view הוא זה שמדפיס את ה - HTML ומחליט איפה למקם את המשתנים.  
  
 - Controllerזה שמתווך בין השניים.

**תרשימי המערכת**

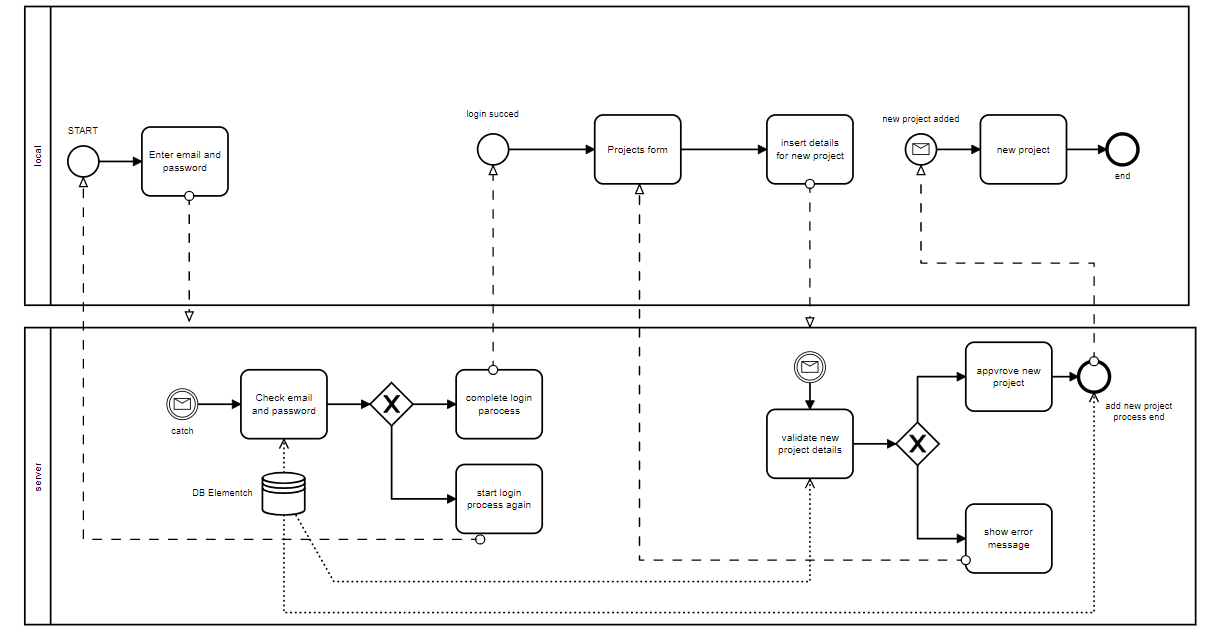
**Use Case - תהליך הוספת משימה חדשה עבור פרויקט והקצאתה למשתמש.**



**Sequence Diagram - תהליך כניסה למערכת**



**Business Process Management - תהליך כניסה 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](https://he.wikipedia.org/wiki/%D7%A9%D7%A4%D7%AA_%D7%AA%D7%9B%D7%A0%D7%95%D7%AA_%D7%93%D7%99%D7%A0%D7%9E%D7%99%D7%AA) [מונחית ־ עצמים](https://he.wikipedia.org/wiki/%D7%AA%D7%9B%D7%A0%D7%95%D7%AA_%D7%9E%D7%95%D7%A0%D7%97%D7%94-%D7%A2%D7%A6%D7%9E%D7%99%D7%9D) המותאמת לשילוב ב[אתרי אינטרנט](https://he.wikipedia.org/wiki/%D7%90%D7%AA%D7%A8_%D7%90%D7%99%D7%A0%D7%98%D7%A8%D7%A0%D7%98) ורצה על ידי [דפדפן האינטרנט](https://he.wikipedia.org/wiki/%D7%93%D7%A4%D7%93%D7%A4%D7%9F_%D7%90%D7%99%D7%A0%D7%98%D7%A8%D7%A0%D7%98) ב[צד הלקוח](https://he.wikipedia.org/wiki/%D7%A6%D7%93_%D7%9C%D7%A7%D7%95%D7%97). |
| OOP | תכנות מונחה עצמים-Programming .Object Oriented |
| MVC | Model-View-Controller [תבנית עיצוב](https://he.wikipedia.org/wiki/%D7%AA%D7%91%D7%A0%D7%99%D7%AA_%D7%A2%D7%99%D7%A6%D7%95%D7%91) ב[הנדסת תוכנה](https://he.wikipedia.org/wiki/%D7%94%D7%A0%D7%93%D7%A1%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) המשמשת ל[הפשטת](https://he.wikipedia.org/wiki/%D7%90%D7%91%D7%A1%D7%98%D7%A8%D7%A7%D7%A6%D7%99%D7%94_(%D7%9E%D7%93%D7%A2%D7%99_%D7%94%D7%9E%D7%97%D7%A9%D7%91)) יישום כלשהו |
| Admin | מנהל המערכת. יכול לגשת לניהול מאגר השאלות על פי סיסמה אישית |
| Web | שיטת תקשורת בין רכיבי תוכנה ב[רשת מחשבים](https://he.wikipedia.org/wiki/%D7%A8%D7%A9%D7%AA_%D7%9E%D7%97%D7%A9%D7%91%D7%99%D7%9D). |
| GUI | הוא [ממשק משתמש](https://he.wikipedia.org/wiki/%D7%9E%D7%9E%D7%A9%D7%A7_%D7%9E%D7%A9%D7%AA%D7%9E%D7%A9) ל[תוכנה](https://he.wikipedia.org/wiki/%D7%AA%D7%95%D7%9B%D7%A0%D7%94) או ל[אתר אינטרנט](https://he.wikipedia.org/wiki/%D7%90%D7%AA%D7%A8_%D7%90%D7%99%D7%A0%D7%98%D7%A8%D7%A0%D7%98), המבוסס על [עיצוב גרפי](https://he.wikipedia.org/wiki/%D7%A2%D7%99%D7%A6%D7%95%D7%91_%D7%92%D7%A8%D7%A4%D7%99) של המסך המוצג למשתמש |
| Design pattern | תבניות עיצוב הן פתרון כללי לבעיה שכיחה בעיצוב תוכנה.  תבנית עיצוב אינה עיצוב סופי שניתן להעבירו הישר ל[קוד](https://he.wikipedia.org/wiki/%D7%A7%D7%95%D7%93_%D7%9E%D7%A7%D7%95%D7%A8), 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 המשמשים 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)**

LOGOUT

**ממשק מנהל**

LOGIN

DASHBOARD

ADMINISTRATOR

CONTACT US

TASKBOARD

CONFIGURAION

USERS

USER GROUP

PROJECTS

TASKS

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**

LOGOUT

LOGIN

DASHBOARD

D

USER PROFILE

TASKBOARD

CONTACT US

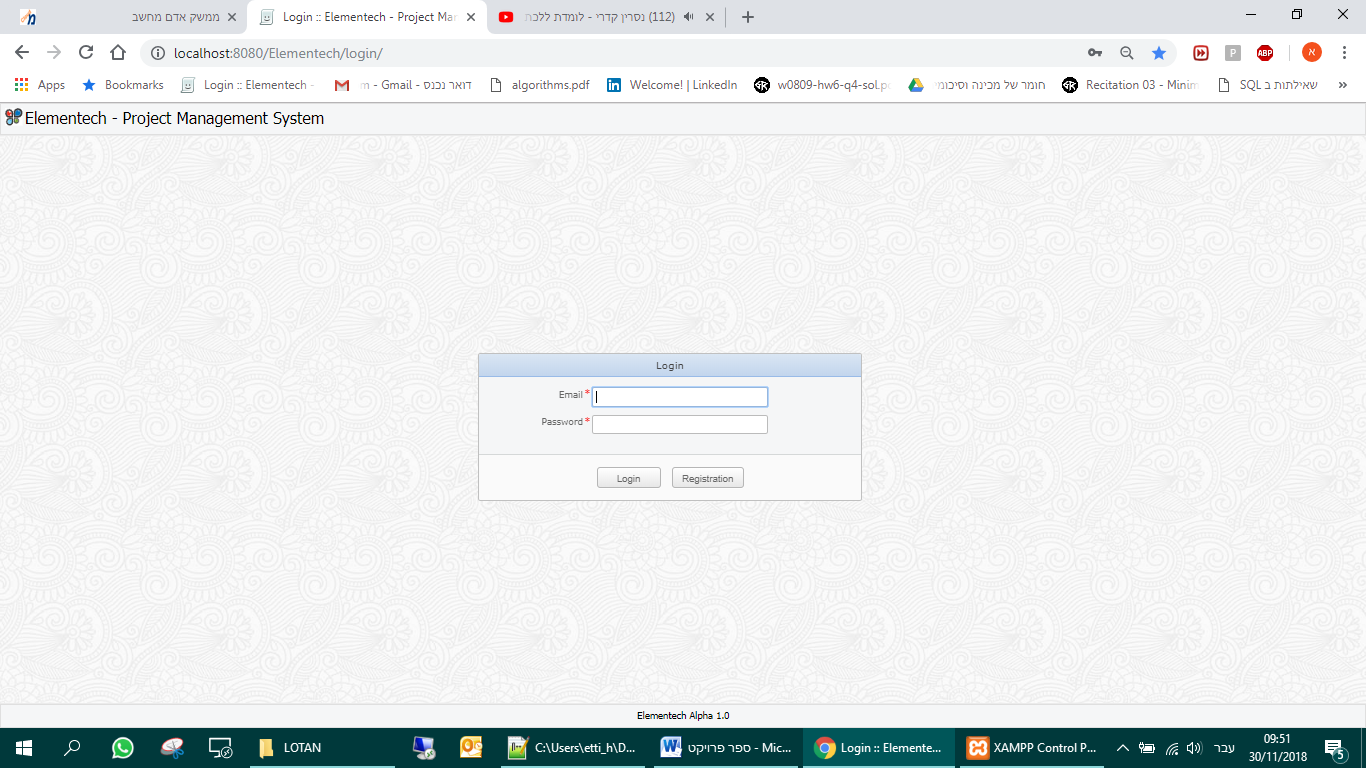
TASKS

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



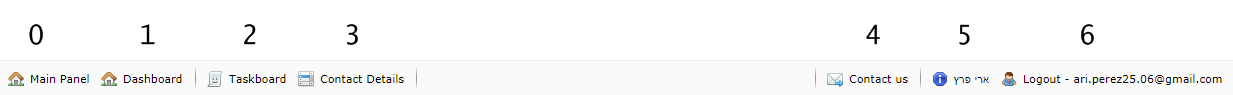
111111

2

**Menu** - תפריט

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

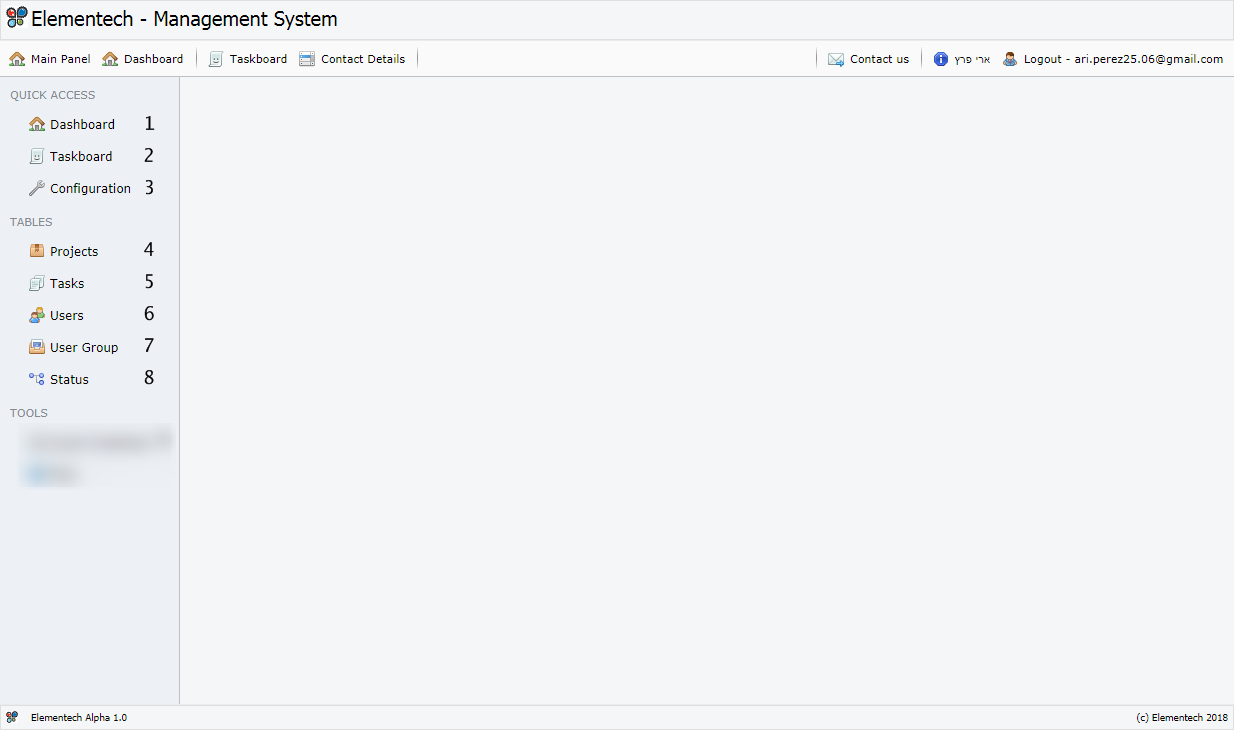
1. Main Panel – פקד שמפנה לתפריט רק עבור אדמין / מנהל / משתמש עם הרשאה בצפיית טבלאות.
2. Dashboard - צפייה בכלל הפעילויות במערכת.
3. Taskboard - צפייה בלוח משימות אישיות.
4. Contact Details – יומן אנשי קשר של כל העובדים.
5. Contact Us - צור קשר.
6. Profile edit – עריכת פרופיל משתמש.
7. בעת לחיצה על פקד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

****

בהמשך נפרט עבור כלל המסכים.

**Main Panel – תפריט עבור אדמין / מנהל**

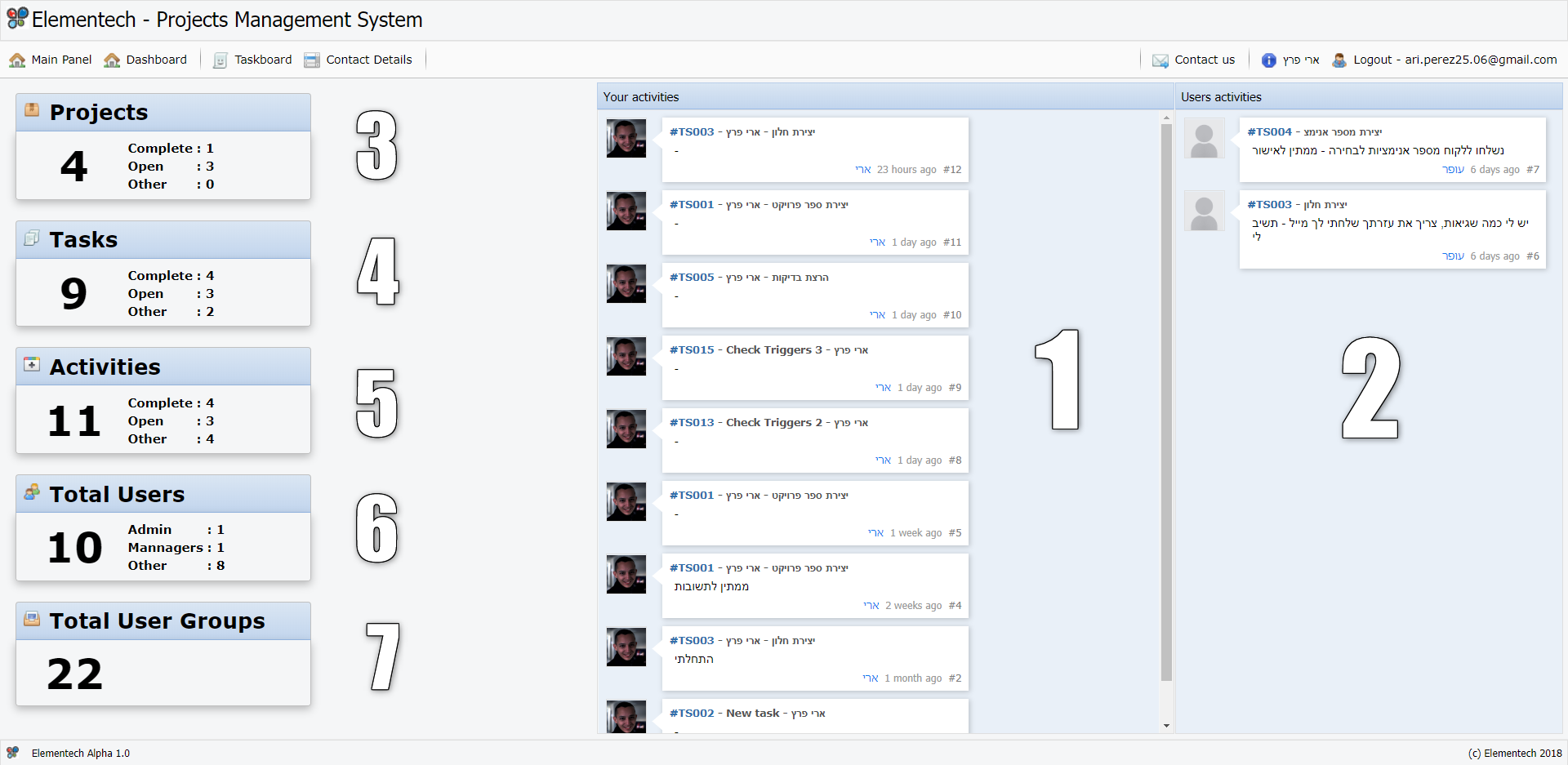
1. Dashboard – צפייה בכלל הפעילויות במערכת.
2. Taskboard – צפייה בלוח משימות אישיות.
3. Configuration – צפייה בדף הגדרות (מוצג רק לאדמין).
4. Projects – צפייה בטבלת פרויקטים (מוצג למנהל רק אם אדמין מאשר זאת).
5. Tasks – צפייה בטבלת משימות (מוצג למנהל רק אם אדמין מאשר זאת).
6. Users – צפייה בטבלת משתמשים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

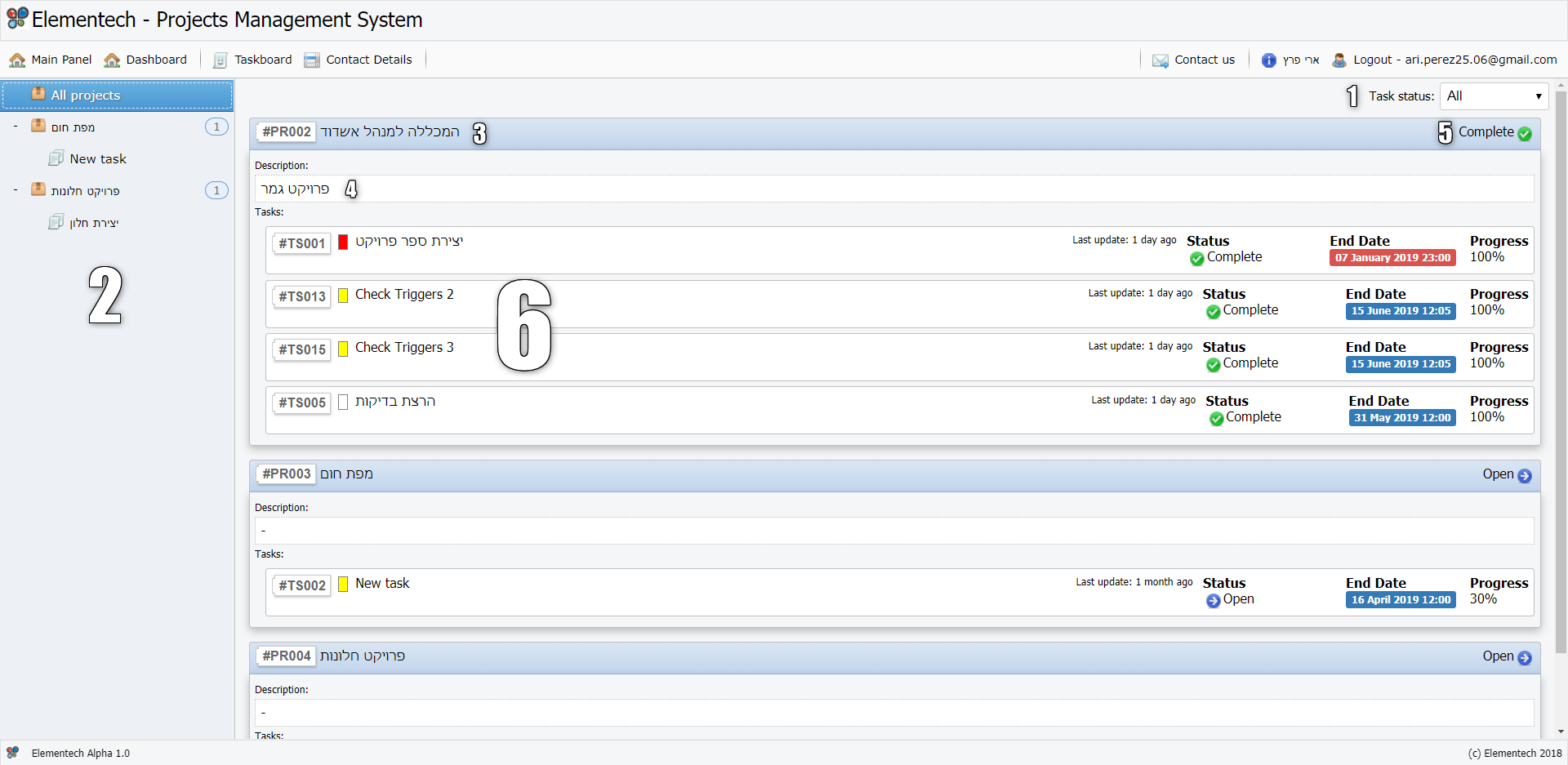
1. עדכוני פעילויות של משתמש.
2. עדכוני פעילויות של משתמשים.
3. סכום כללי של פרויקטים.
4. סכום כללי של משימות.
5. סכום כללי של פעילויות.
6. סכום כללי של משתמשים (מוצג רק עבור אדמין).
7. סכום כללי של מחלקות (מוצג רק עבור אדמין).

****

**Taskboard – לוח משימות**

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

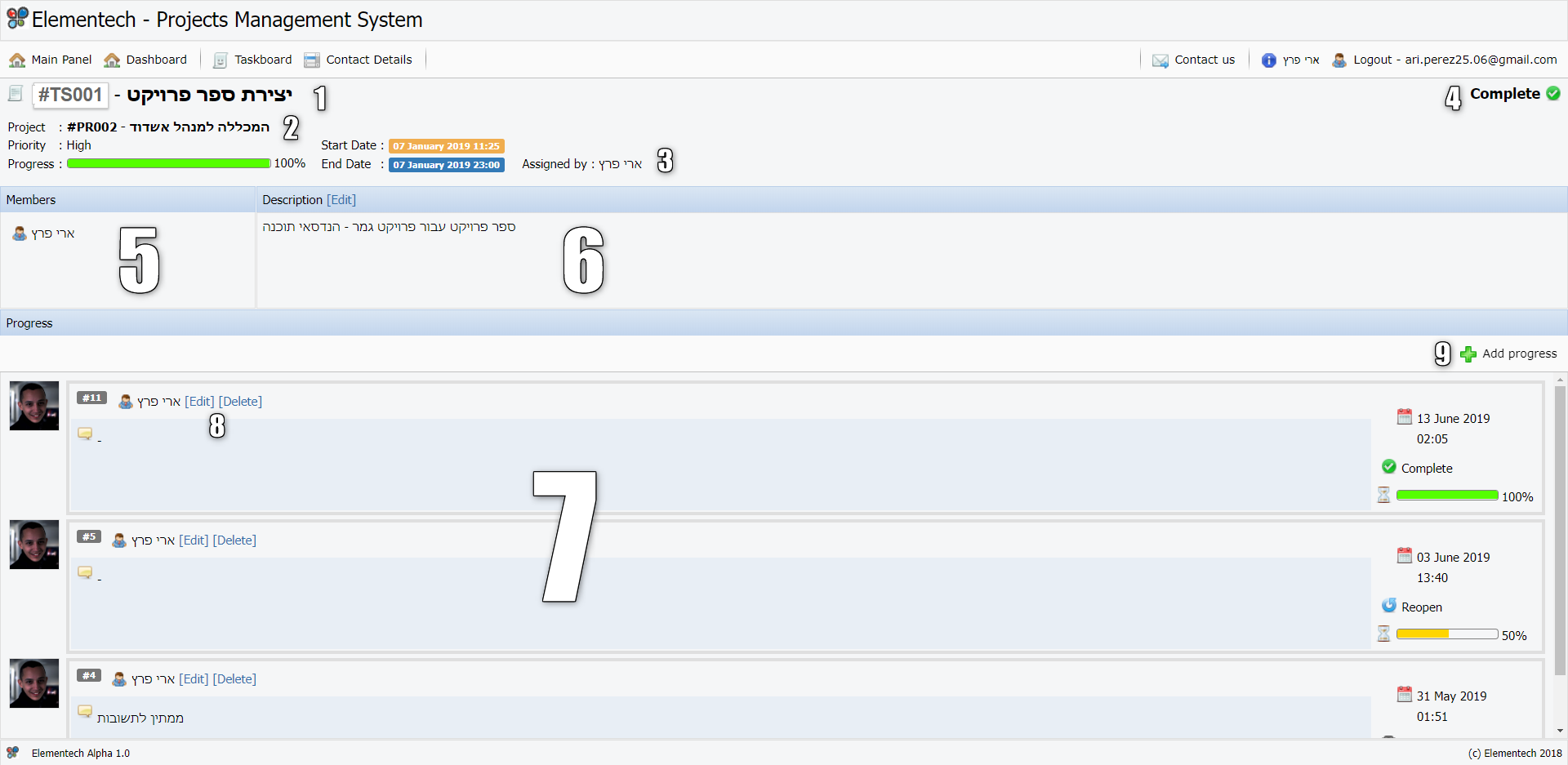
1. פקד רשימה נפתחת – אפשרות זו נותנת להציג את כל המשימות עבור אותו משתמש, ואת כל המשימות שהושלמו.
2. חלק זה במסך מציג את המשימות שהמשתמש קיבל, כל משימה נמצאת תחת שם הפרויקט המתאים שלה.
3. שם הפרויקט.
4. תיאור הפרויקט.
5. סטטוס פרויקט.
6. משימות הקשורות לאותו פרויקט.

****

לחיצה על מספר המשימה (פקד 1) תוביל למסך הבא, שבו ניתן להוסיף עדכונים עבור המשימה כדי שמשתמשים 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.

****

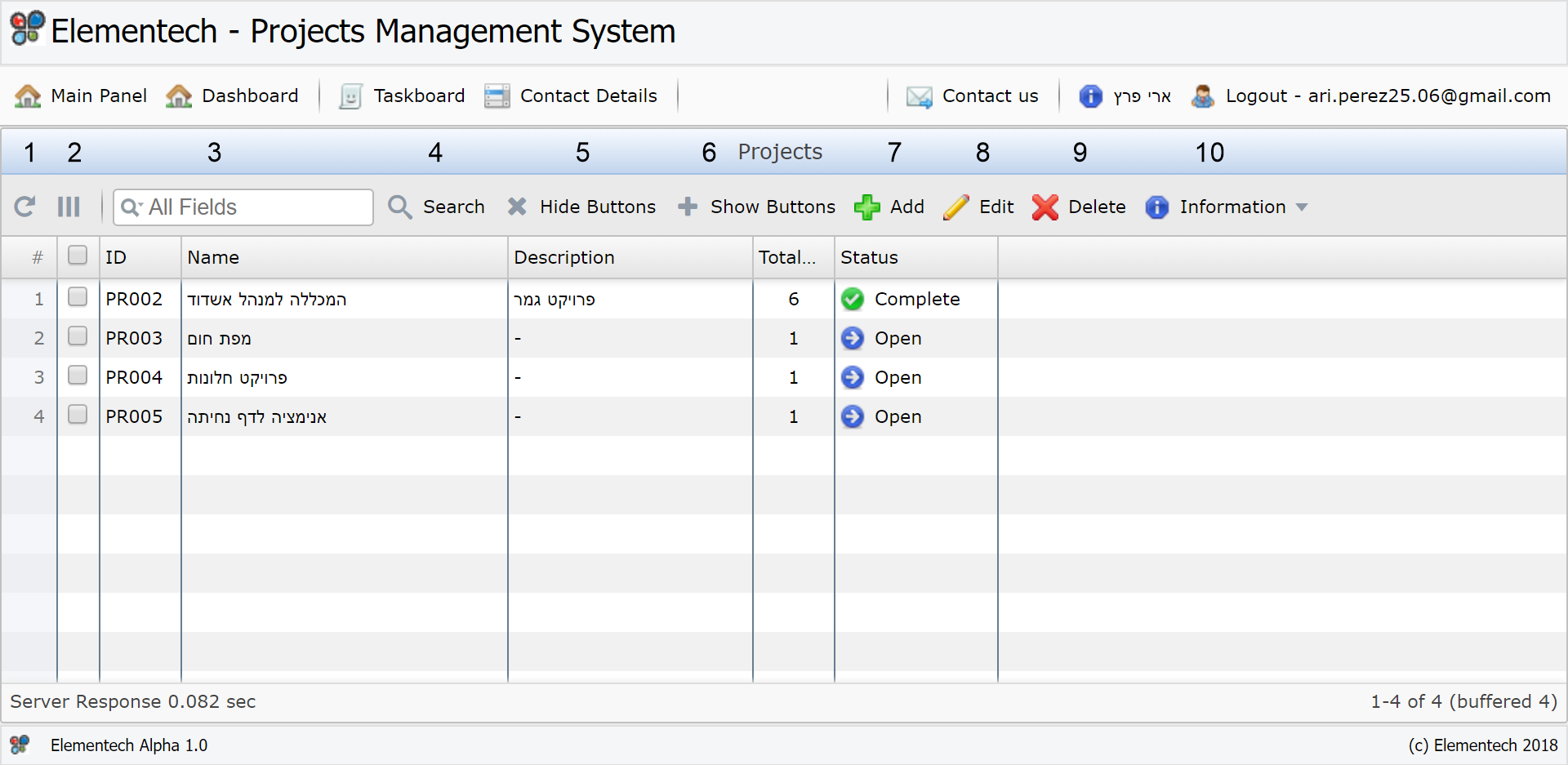
**Projects - פרויקטים**

**מסך לניהול פרויקטים**

1. ריענון דף.
2. מיון השדות במסך – ניתן להעלים שדות במסך לצורך השוואה או חוסר טעם.

או לשמור על התצורה האחרונה של הטבלה ע"פ בחירת המשתמש.

1. חיפוש בטבלה.
2. חיפוש לפי פרמטרים.
3. העלמת כפתורי הוספה \ עריכה \ מחיקה.
4. החזרת כפתורי הוספה \ עריכה \ מחיקה.
5. הוספת פרויקט חדש.
6. עריכת פרויקט קיים
7. מחיקת פרויקט.
8. מידע על המסך.

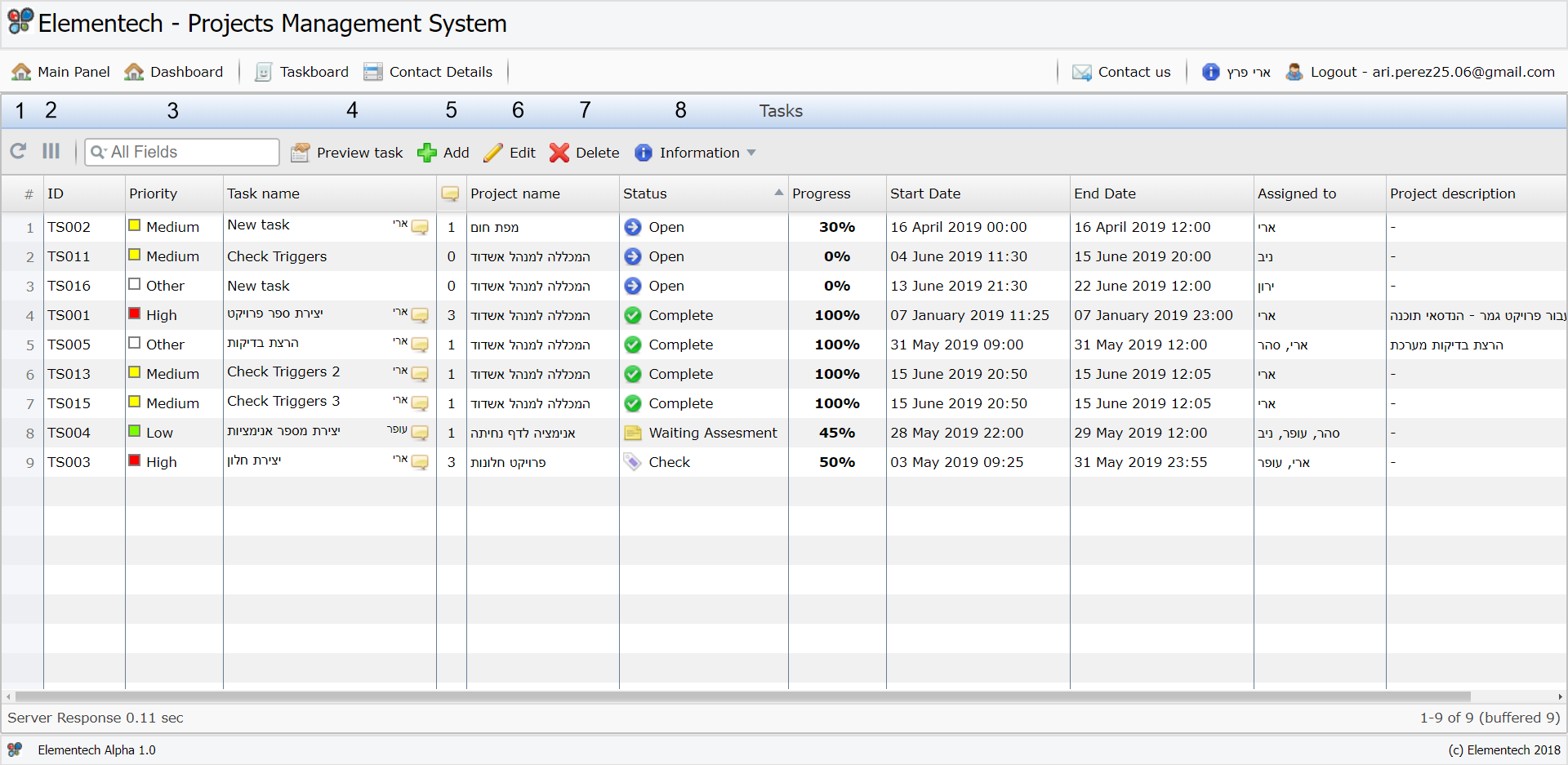


**Tasks - ניהול משימות**

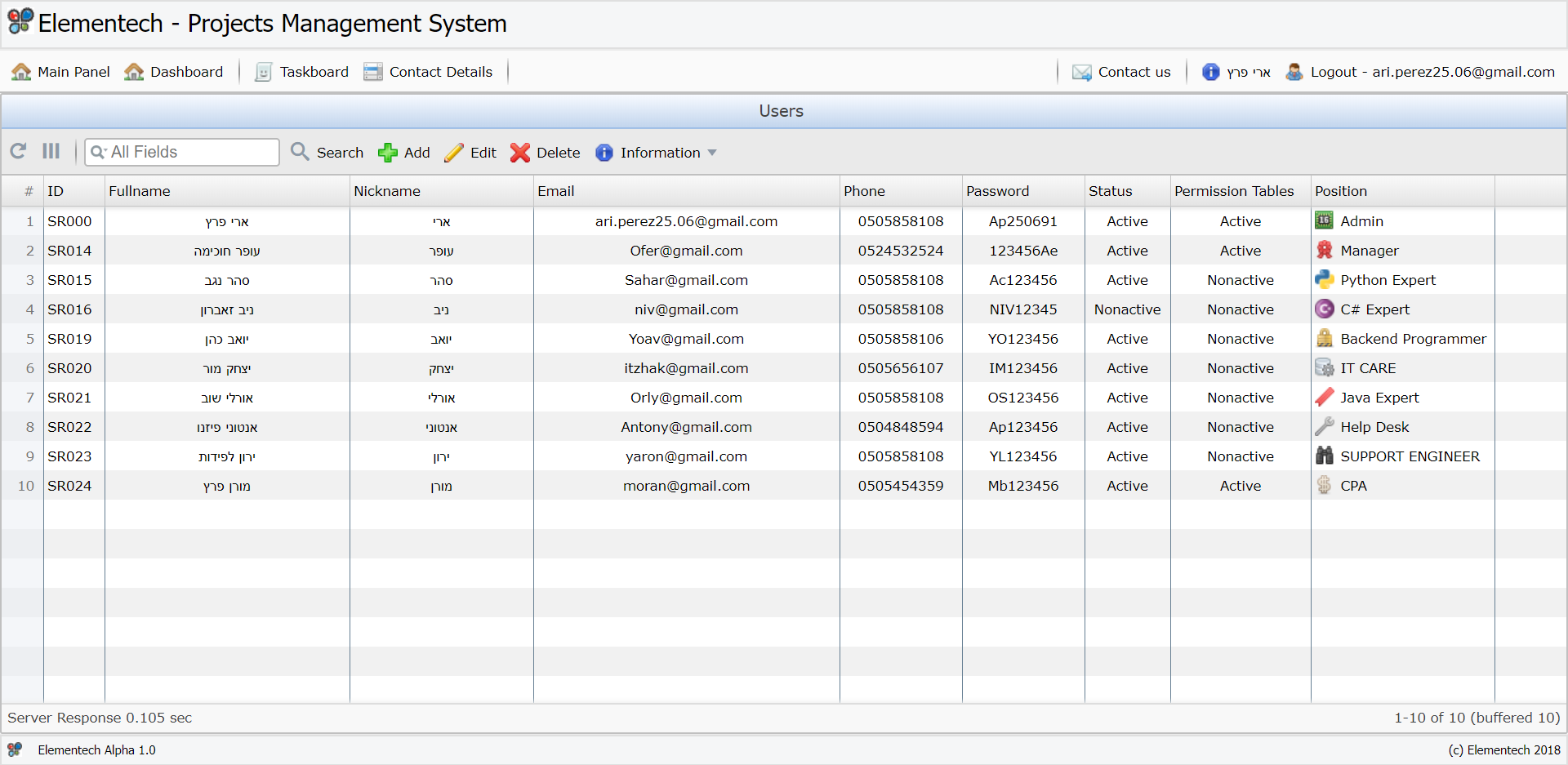
1. ריענון דף.
2. מיון השדות במסך – ניתן להעלים שדות במסך לצורך השוואה או חוסר טעם.

או לשמור על התצורה האחרונה של הטבלה ע"פ בחירת המשתמש.

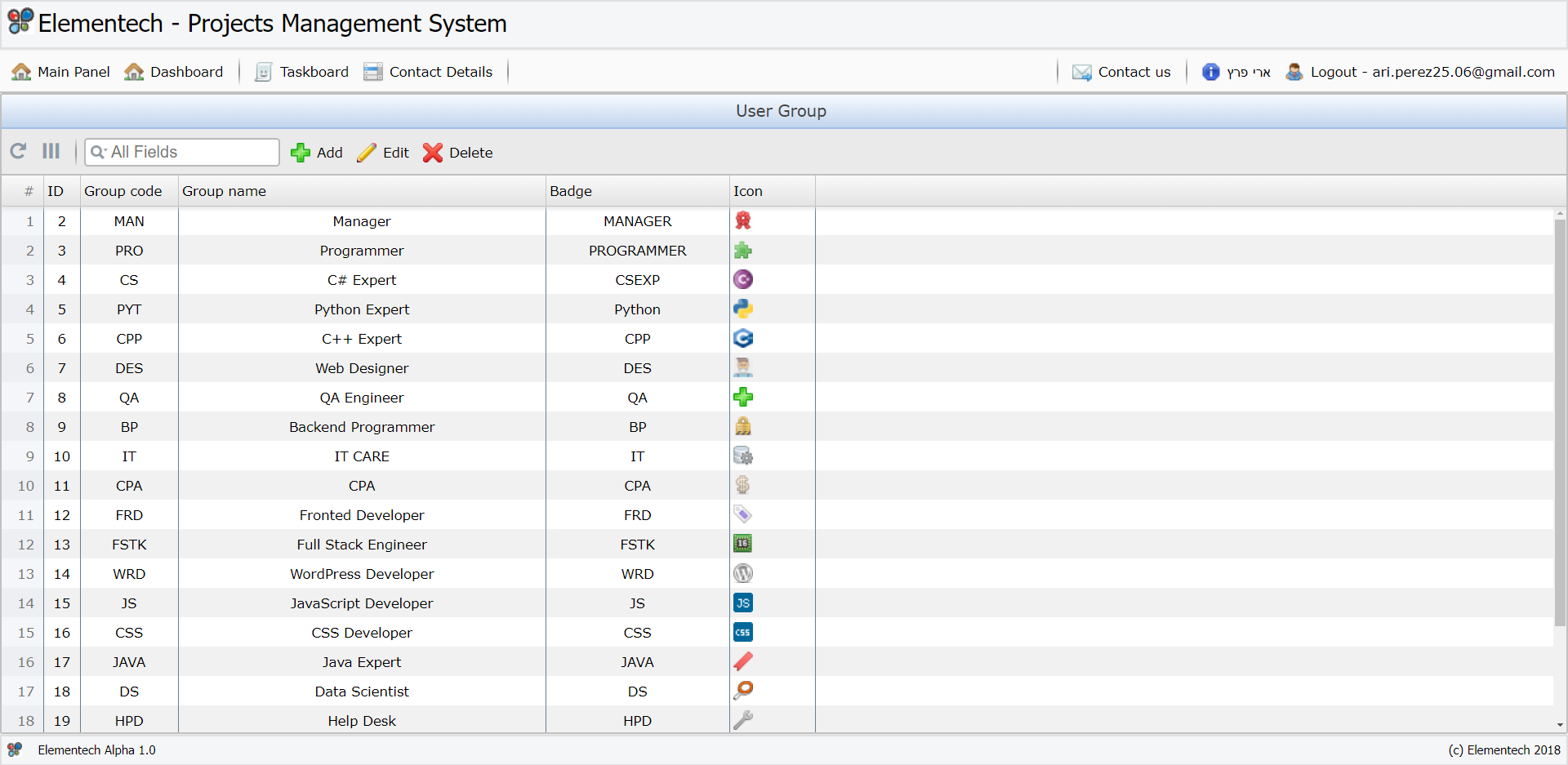
1. חיפוש לפי פרמטרים.
2. מעבר למסך פרטי משימה (ניתן גם ע"י דאבל קליק על משימה מסוימת בטבלה).
3. הוספת משימה חדשה.
4. עריכת משימה קיימת.
5. מחיקת משימה.
6. מידע על המסך.

****

**Users - ניהול משתמשים**

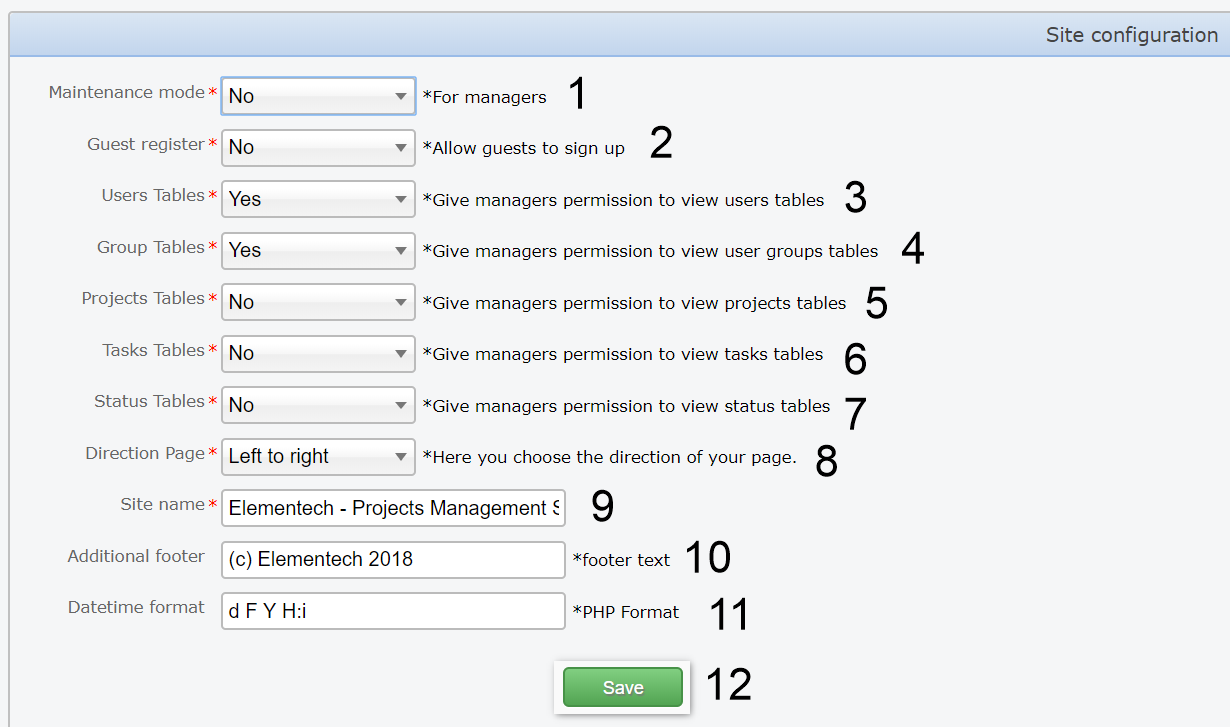
****

**User Group - ניהול קבוצות משתמשים (מחלקות)**

****

**Configuration - קונפיגורציית האתר**

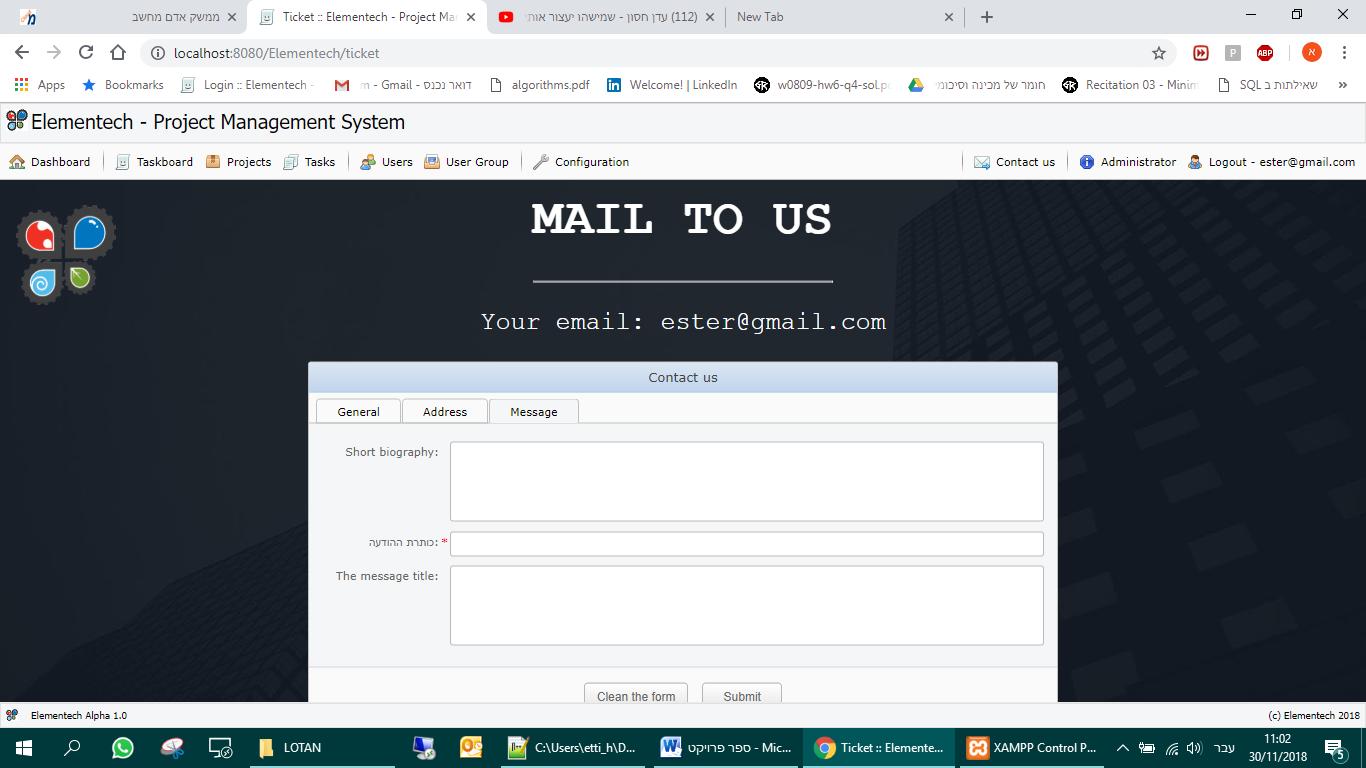
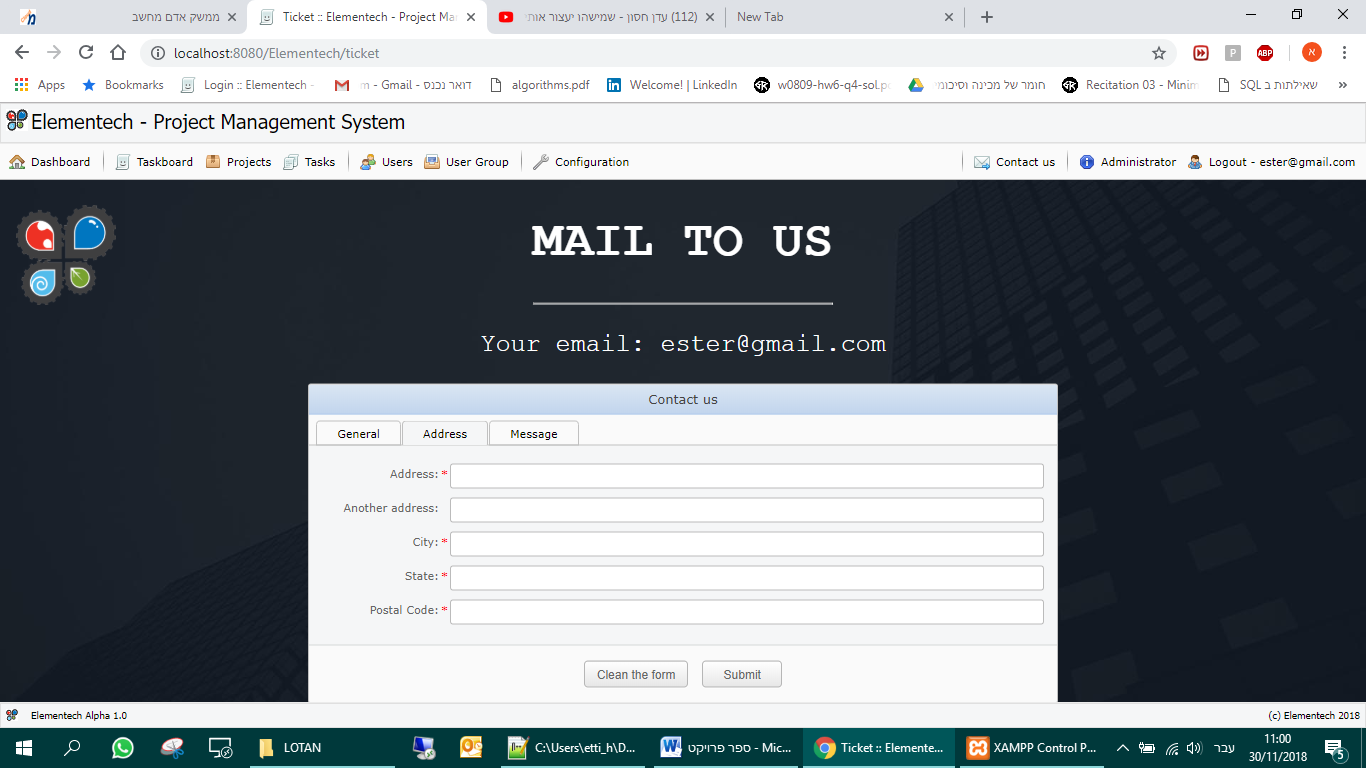
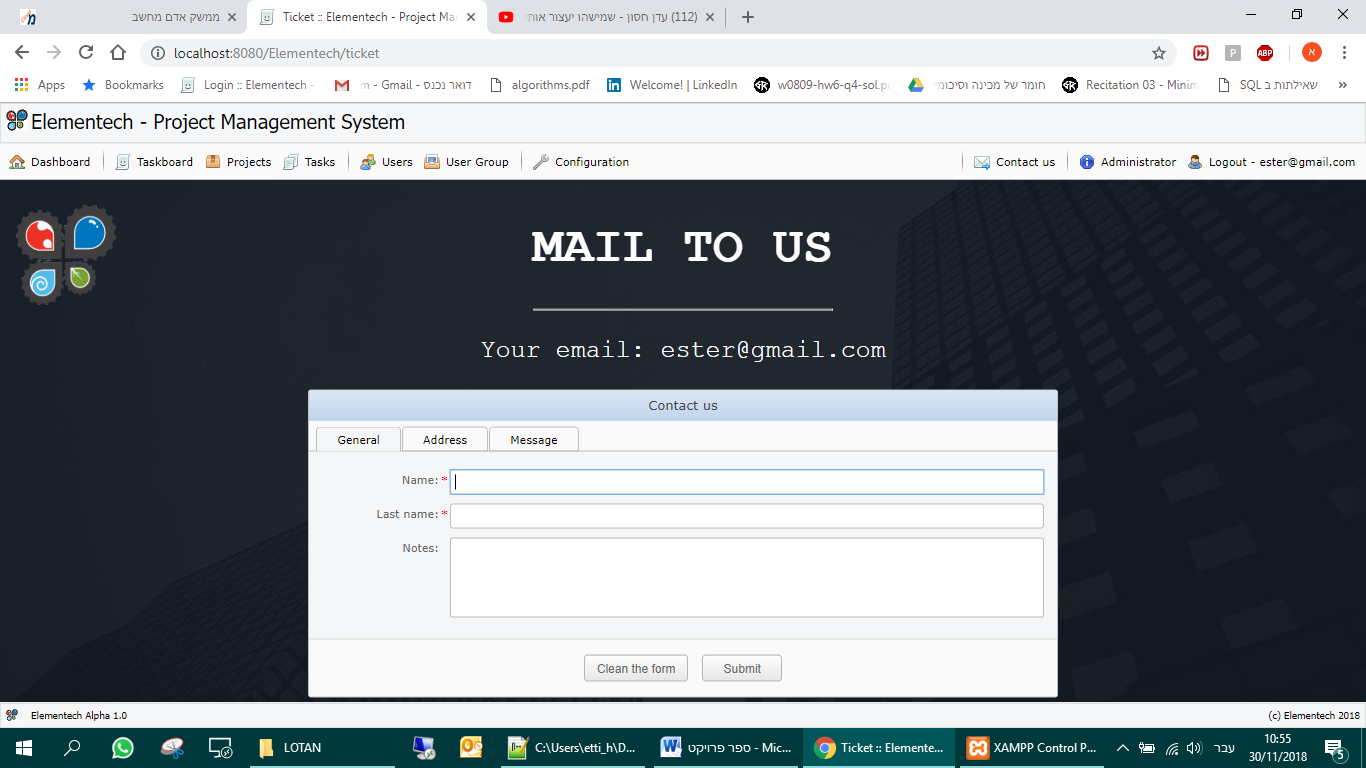
1. מציין כי המערכת במצב תחזוקה כך שמשתמשים יהיו מודעים לכך אם במידה ויש קריסות בזמן העבודה.
2. הרשאת הרשמה עבור משתמשים חדשים (מוסיף פקד REGISTER במסך ה-LOGIN).
3. הרשאת צפייה בטבלת משתמשים.
4. הרשאת צפייה בטבלת קבוצות משתמשים.
5. הרשאת צפייה בטבלת פרויקטים.
6. הרשאת צפייה בטבלת משימות.
7. הרשאת צפייה 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).

****

**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.



511111

411111

311111

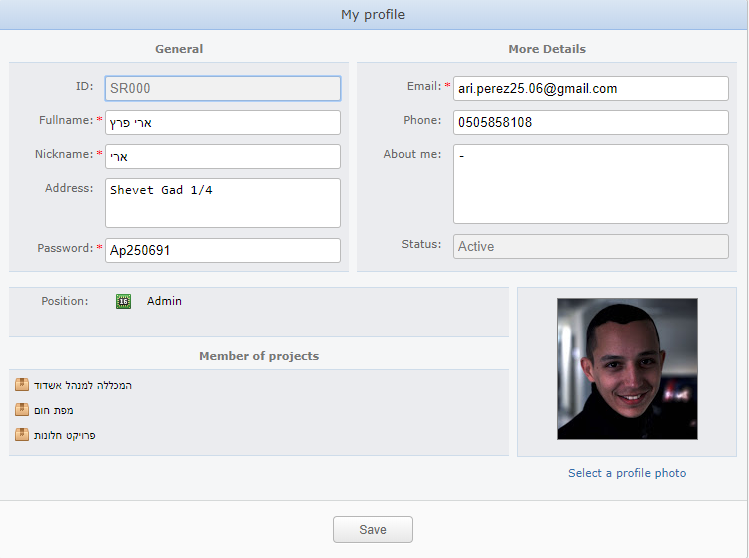
211111

111111

**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.

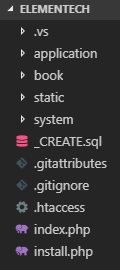


# 4. מבנה נתונים וארגון הקבצים

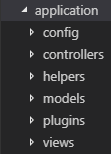
## 4.1 ארגון קבצים

### 4.1.1 כללי

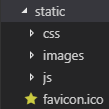
בתוך תיקיית הפרויקט קיים קובץ DB שבעצם מקים את בסיס הנתונים.



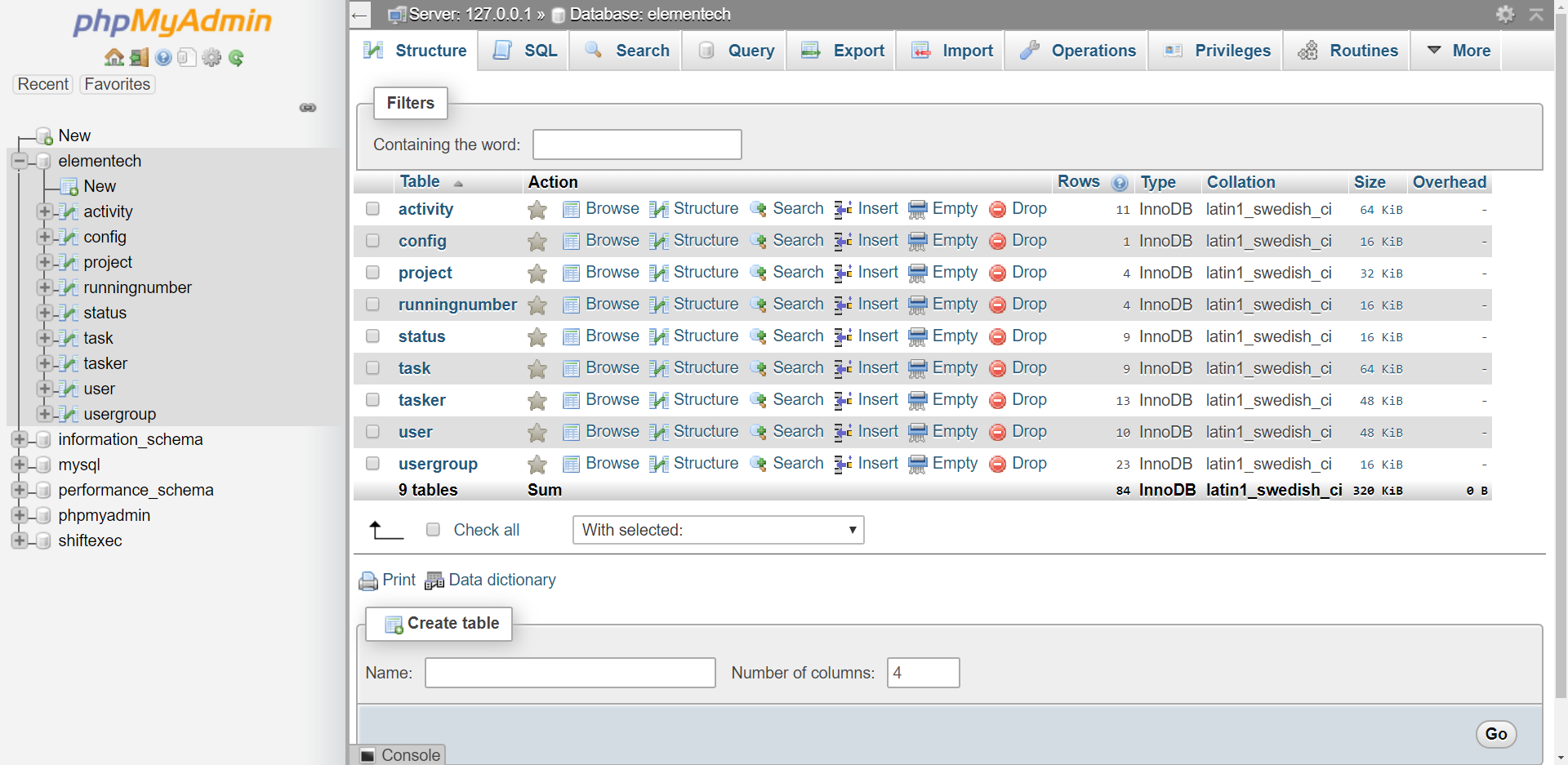
בתוך תיקיית ה-application קיימות תיקיות הפרויקט, controllers, models, views אשר מכילות את מסכי המערכת, לוגיקה, פונקציות שונות במערכת.



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



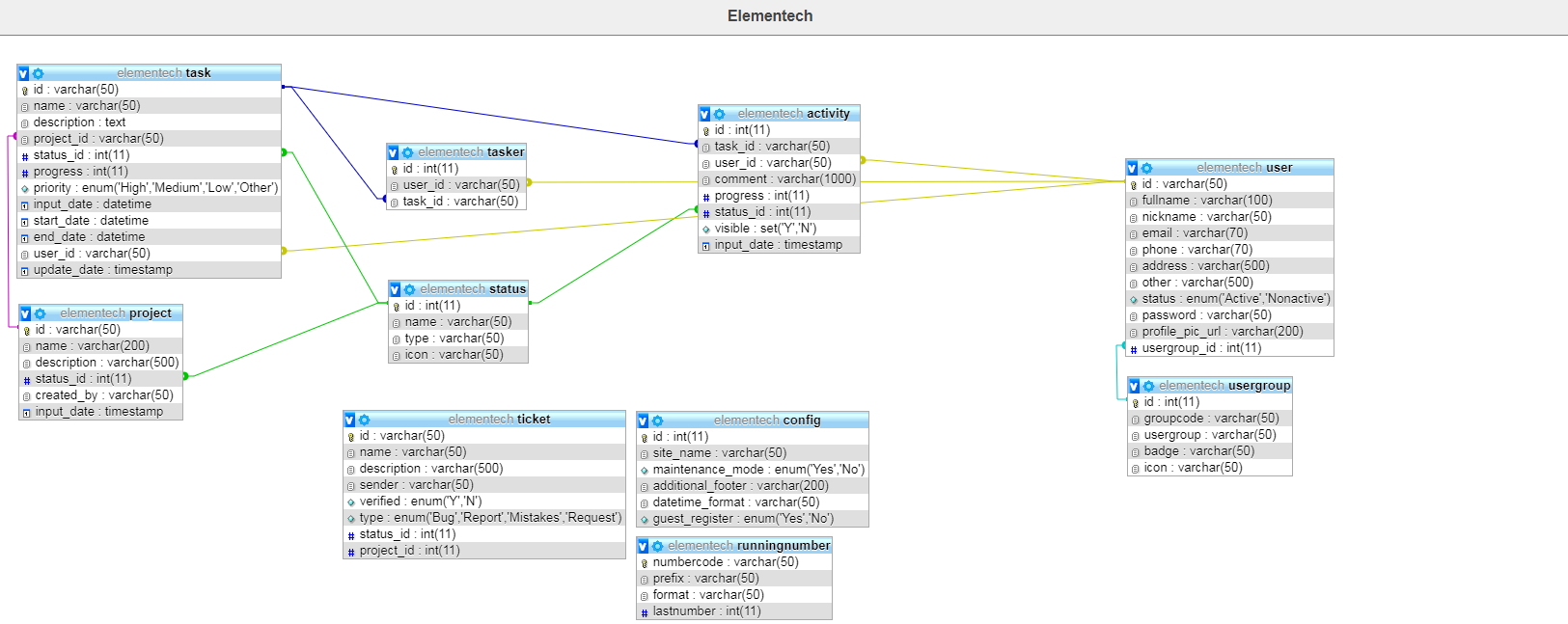
בסיס הנתונים מורכב מטבלאות שונות, והקשרים בינהם מוצגים ב-ERD הבא.

****

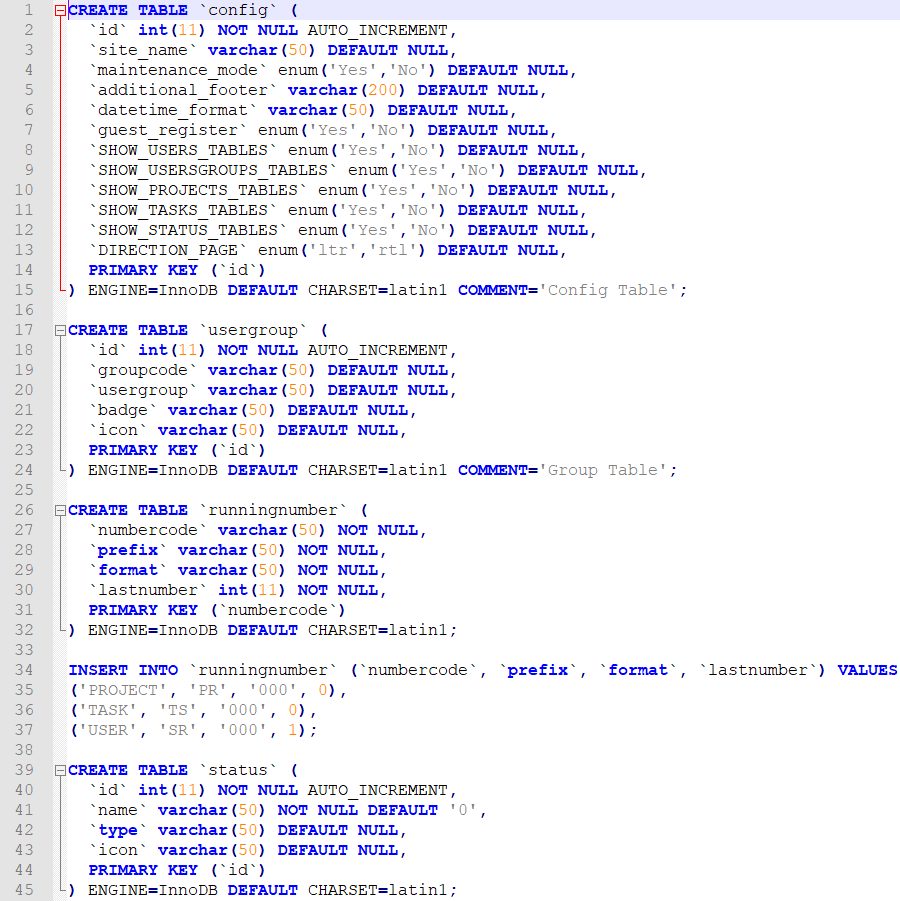
## 4.2 מבנה נתונים

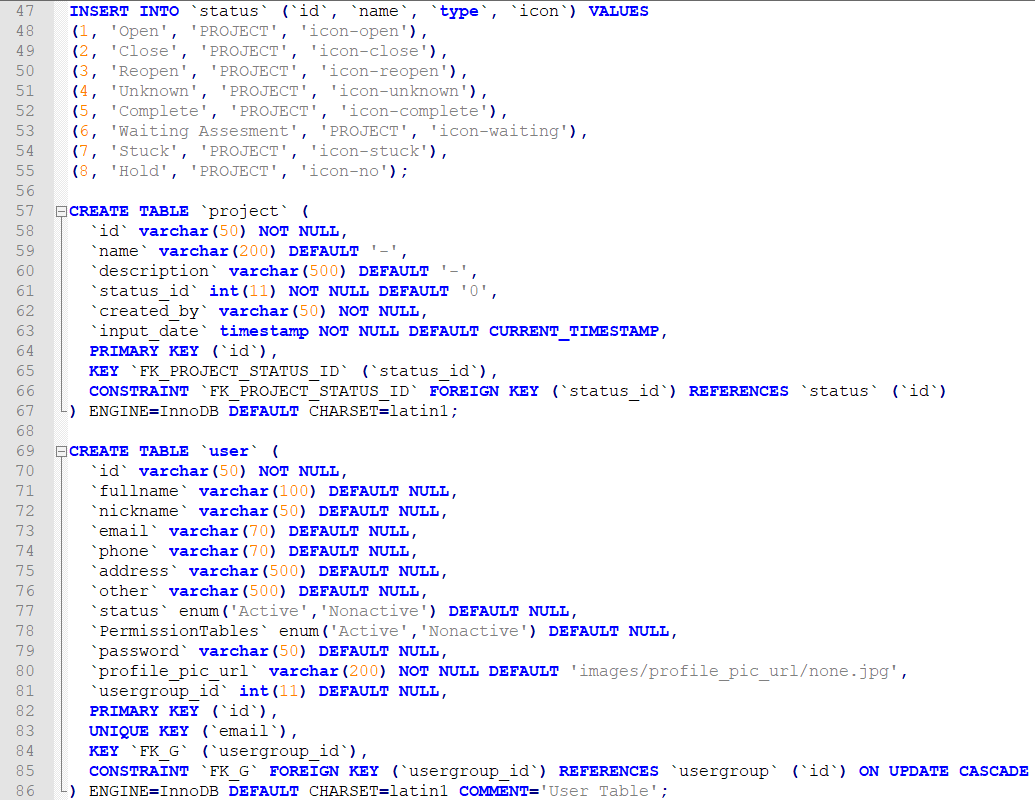
**תרשים ERD של בסיס הנתונים שמכיל נתונים אודות משתמשים, פרויקטים, משימות ועוד.**

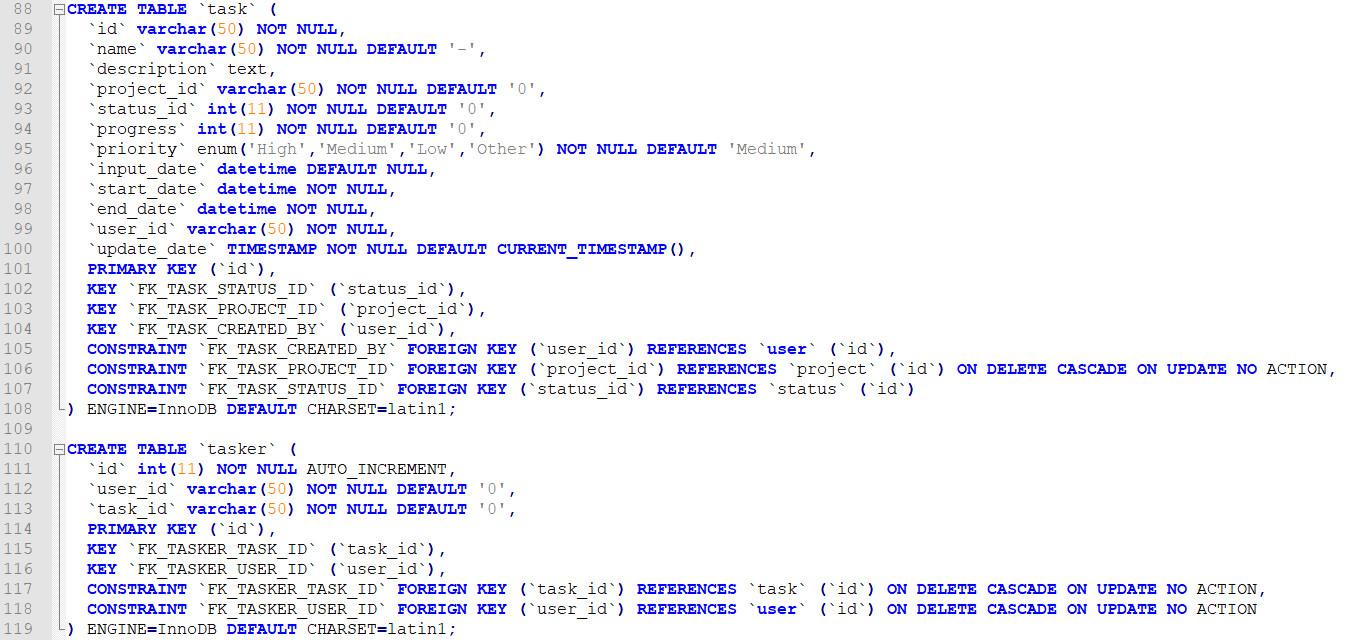
* **טבלת Task** – מספר משימה, שם משימה, תיאור, התקדמות, עדיפות (High, Medium, Low), תאריך הקלט, תאריך התחלה, תאריך סיום, תאריך עדכון, מספר משתמש (FK), מספר פרויקט (FK), סטטוס (FK).
* **טבלת Activity**  - מספר פעילות, הערה, התקדמות, נראה (Y\N), תאריך הקלט, מספר משימה (FK), מספר משתמש (FK), סטטוס (FK).
* **טבלת Config** - מספר הקונפיגורציה, שם האתר, תחזוקה (Y\N), הוספת footer , פורמט התאריך, אורח (Y\N), הצג טבלת משתמשים (Y\N), הצג טבלת קבוצות משתמשים (Y\N), הצג טבלת פרויקטים (Y\N), הצג טבלת סטטוסים (Y\N), שנה כיוון מסכים (ltr\rtl).
* **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.

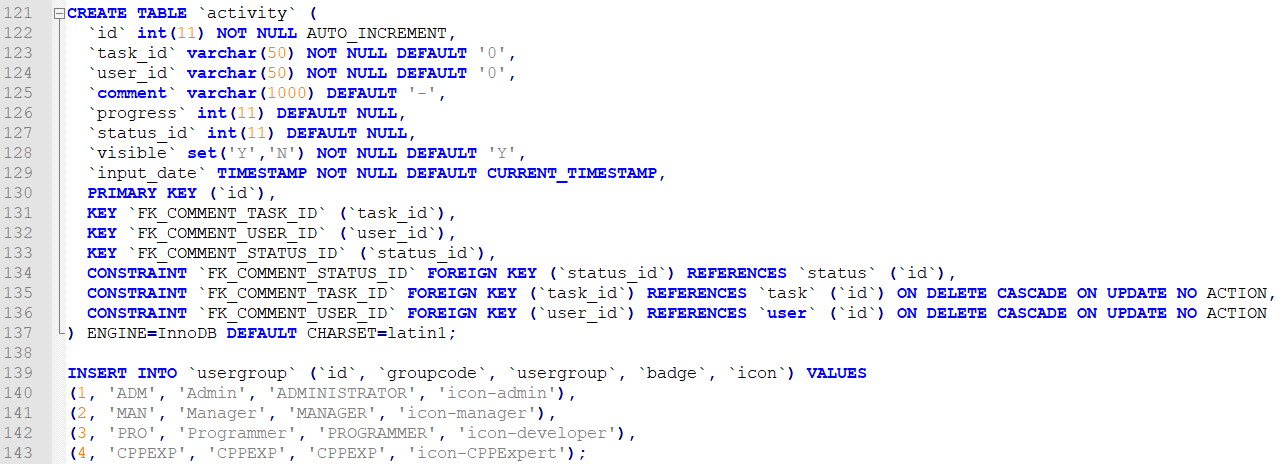


**קטעי קוד - הגדרות מבני נתונים**

****

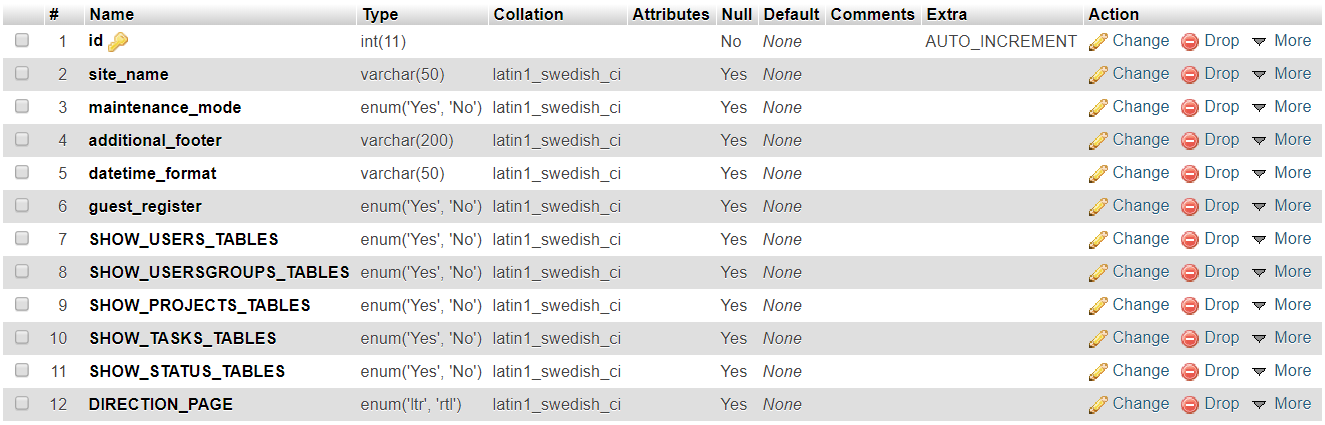




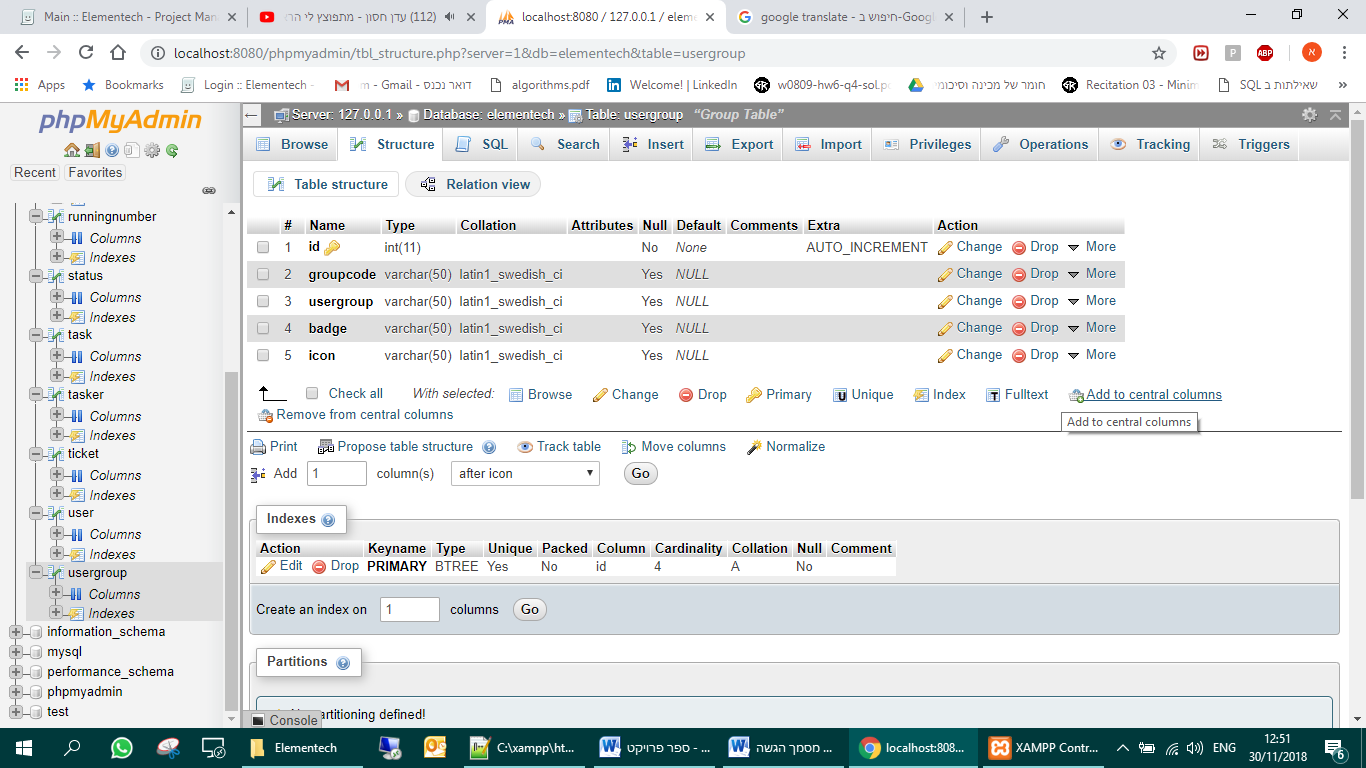
****

**צילומי מסך של הטבלאות מתוך קטעי הקוד המתוארים מעלה**

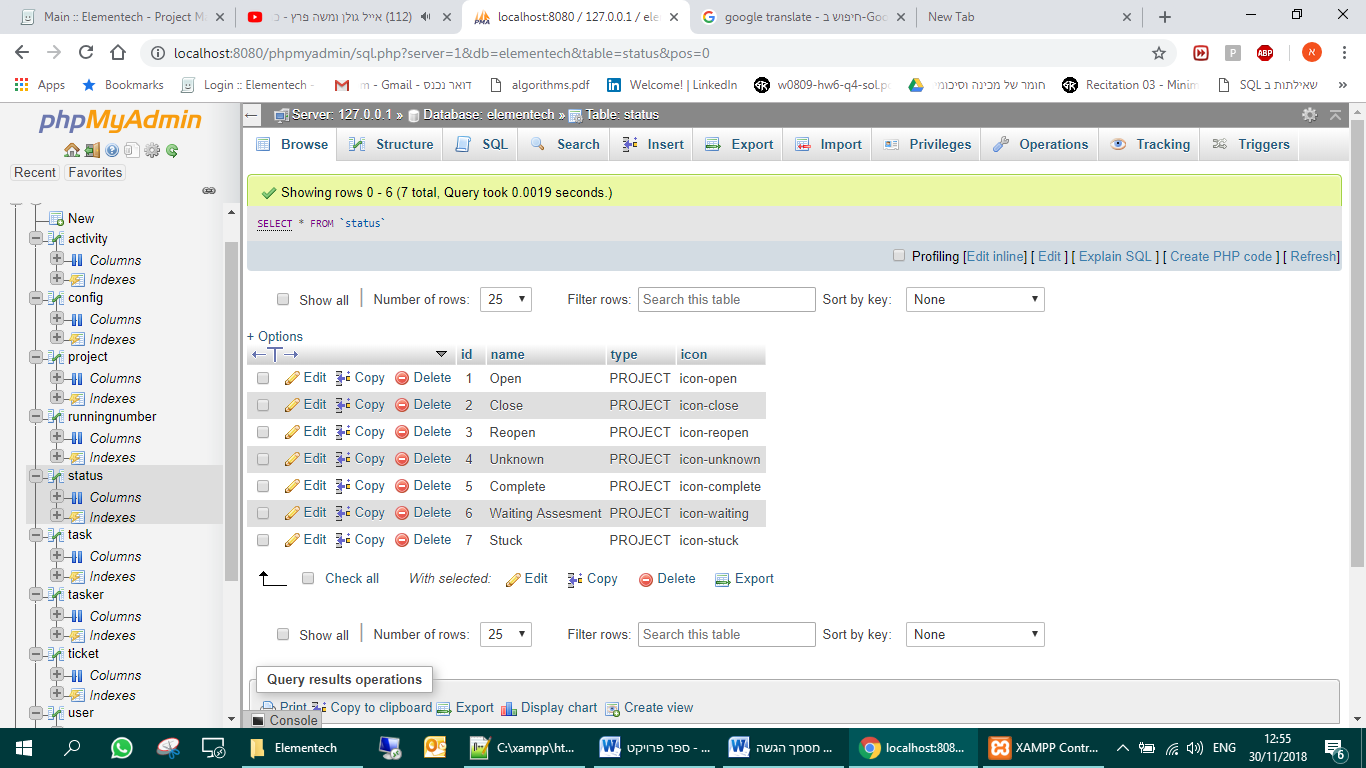
**Config table**



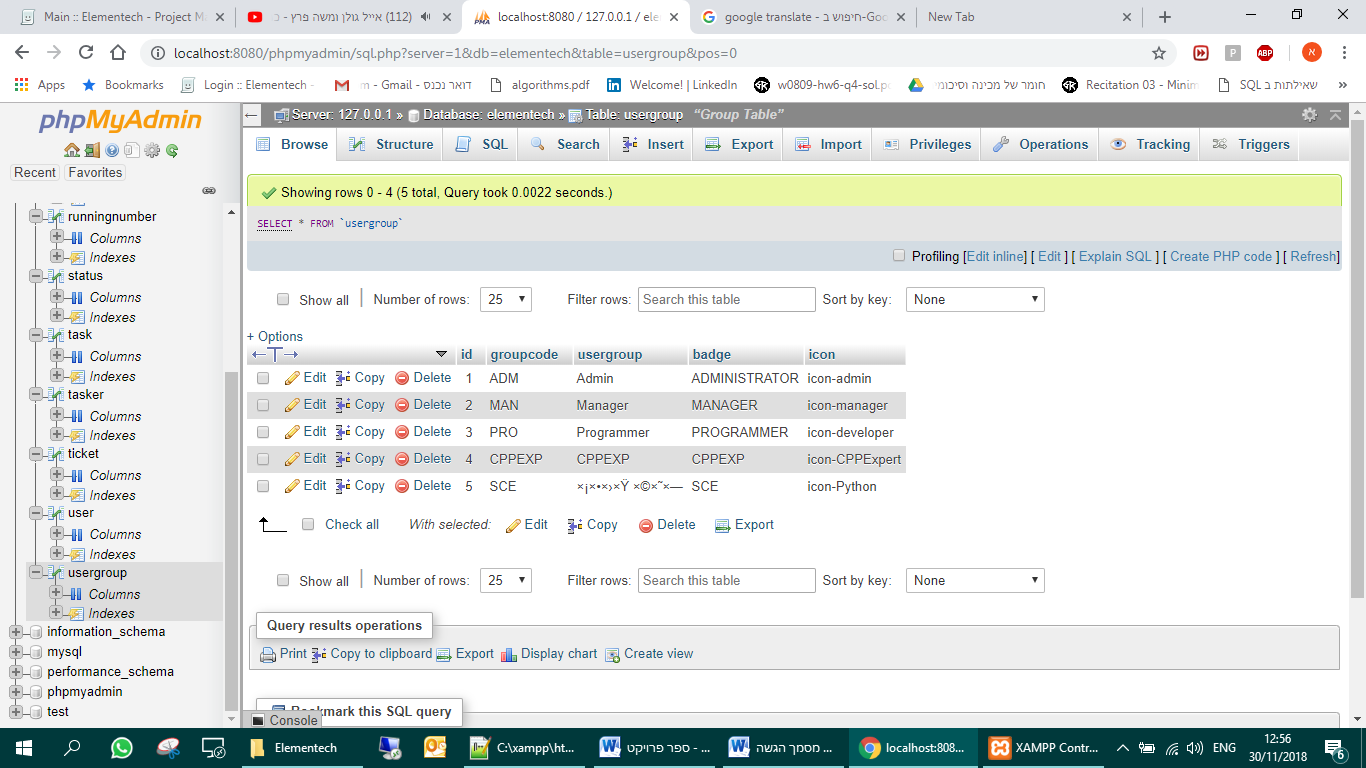
**UserGroup Table**



**Status Table after insert**



**User Group Table after insert**



# 5. תכנון

## 5.1 כללי ועקרונות התכנות (5.2)

Elementech היא מערכת ניהול פרויקטים מבוססת אינטרנט.

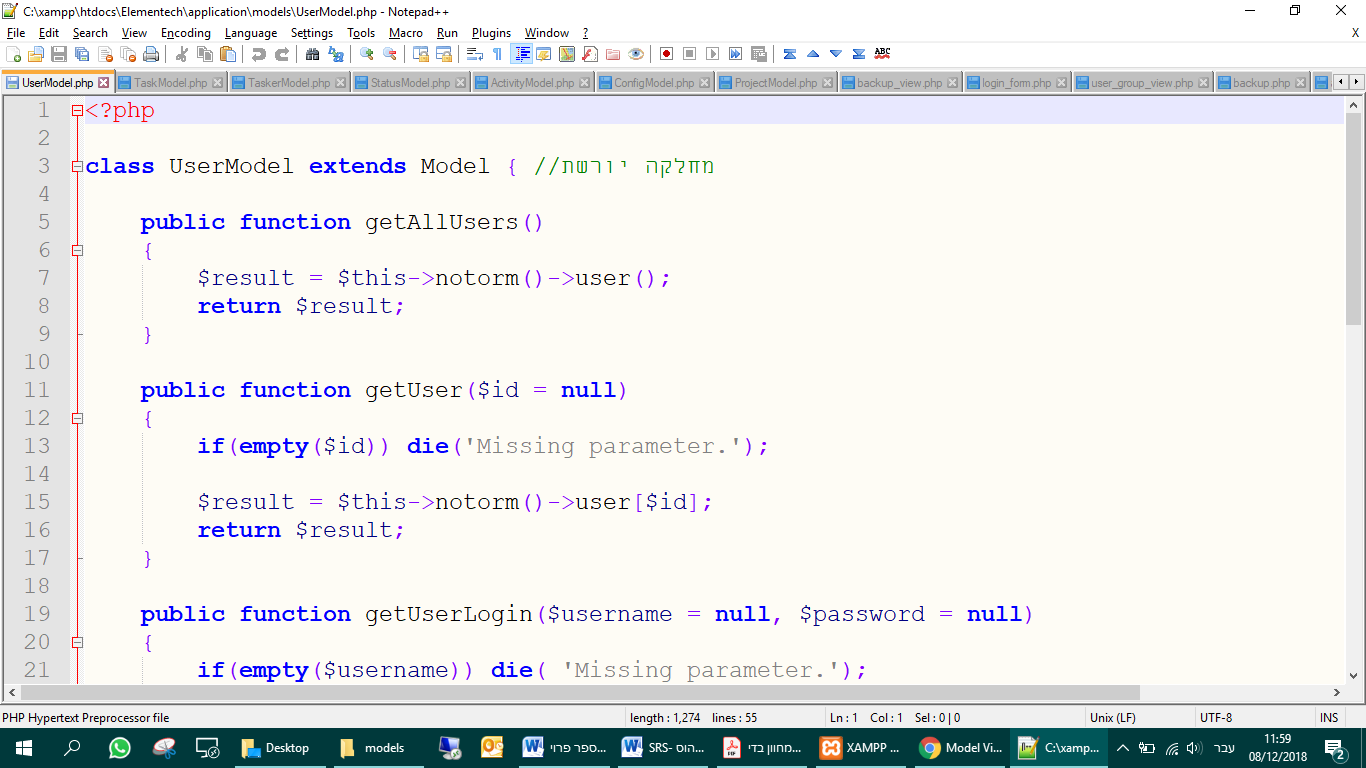
המערכת נכתבה בשפת PHP עם שימוש בטכנולוגיית ajax. השתמשנו בבסיס נתונים MySQL ולצורך עבודה עם בסיס הנתונים השתמשנו ב-jQuery, שהיא תוכנת קוד פתוח שאפשרה לנו ליצור אנימציות ואפקטים, לטפל באירועים במהלך שימוש המשתמש. בנוסף תוכנה זו משתמשת לפיתוח יישומי Ajax. כדי ליצור ממשק משתמש נוח, פשוט וידידותי, השתמשנו 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.



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 Taskboardקיימת 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**

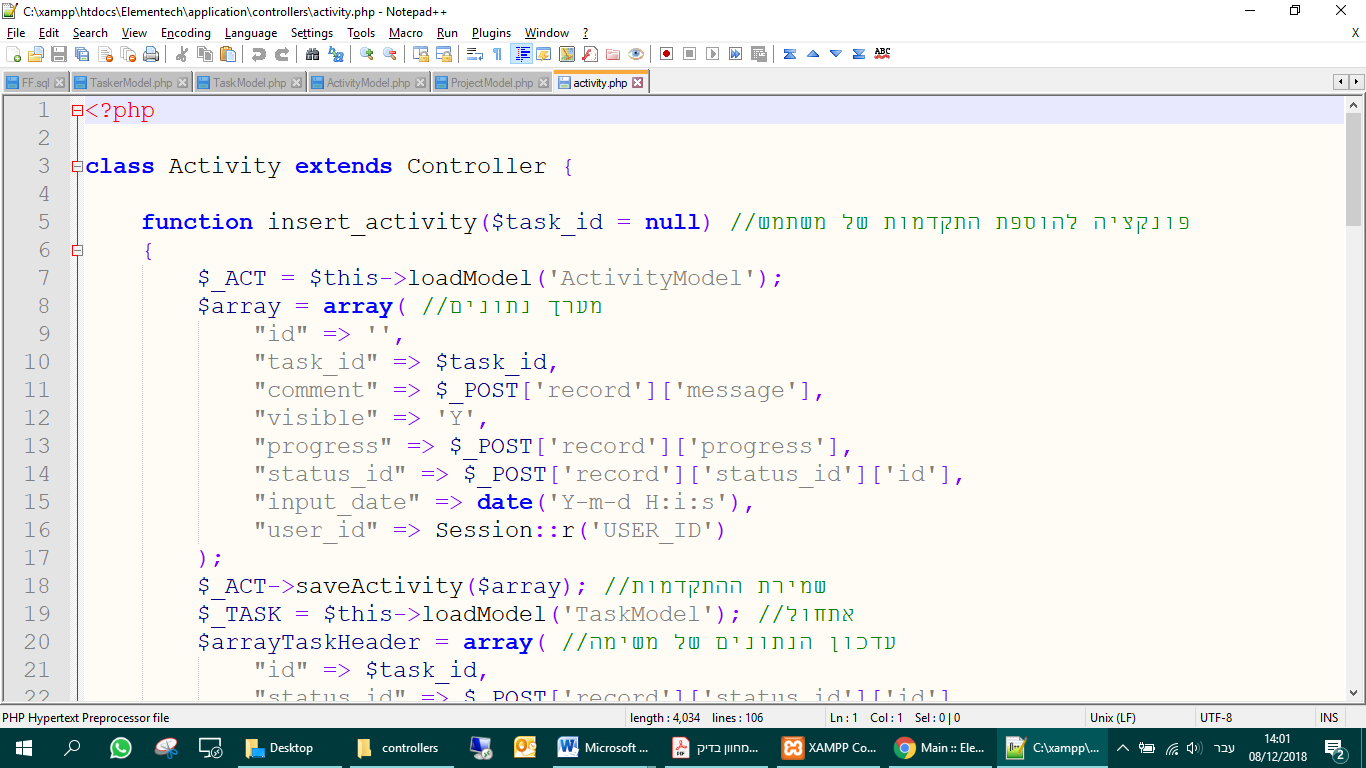
* הוספת פרויקט addProject(array)
* שמירת פרויקט saveProject(array)
* הוספת קבוצת משתמשים addUserGroup(array)
* הוספת משתמש addUser(array)
* הוספת פעילות insert\_activity(task\_id)
* הוספת משימה go\_insert\_task(id)
* הוספת משתמש חדש go\_insert\_user()
* הרשמה למערכת register()

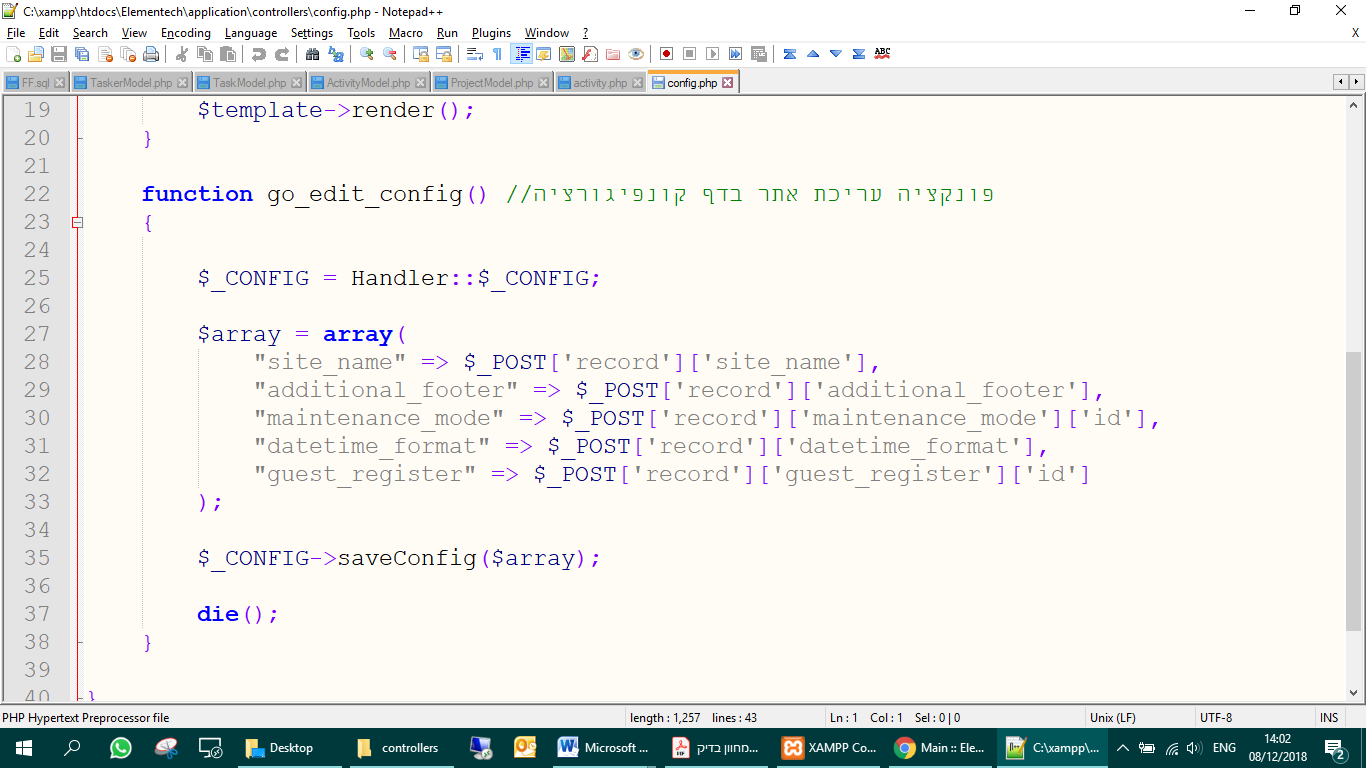
**פונקציית עדכון \ מחיקה**

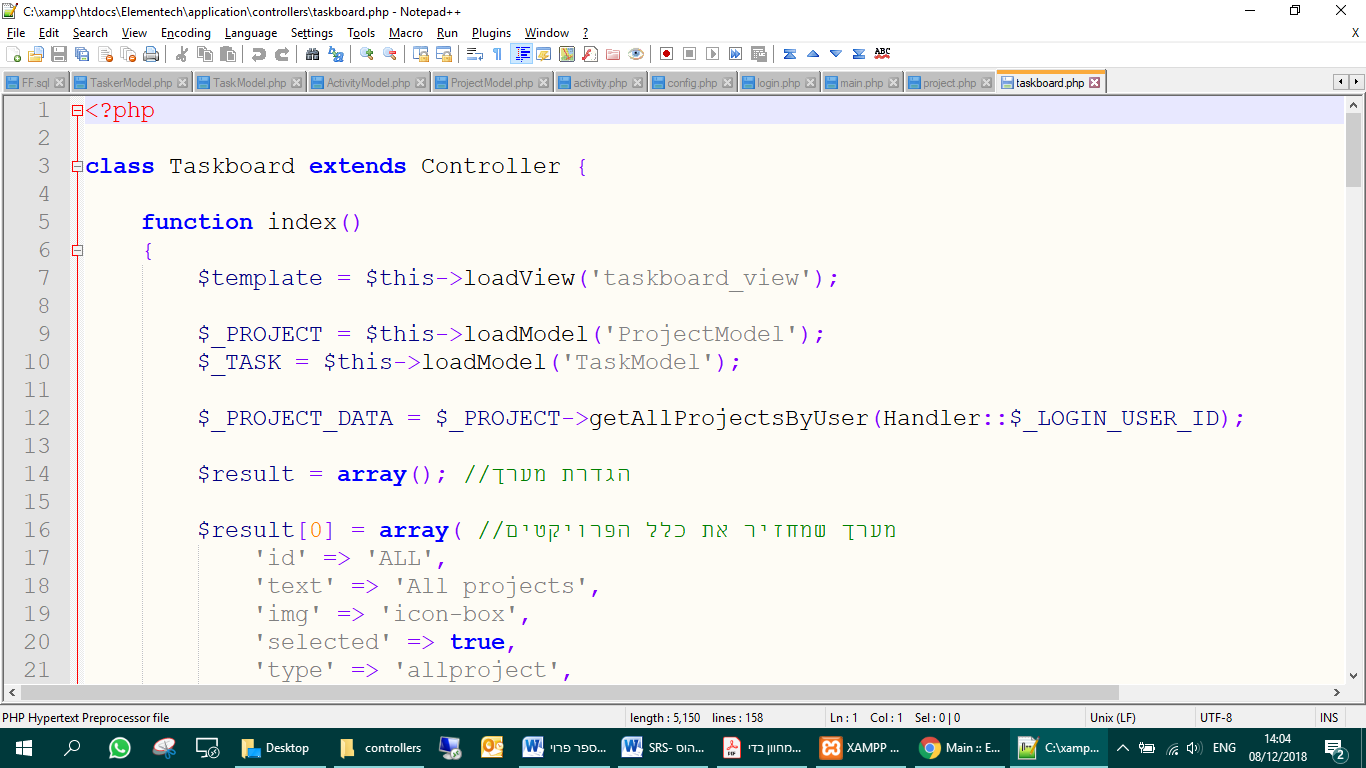
* עדכון משתמש go\_edit\_user(id)
* מחיקת משתמש delete\_user(id)
* עדכון פרופיל go\_edit\_profile(id)
* העלאת תמונת פרופיל upload\_profile\_pic(user\_id)
* עדכון משתמשי קבוצה go\_edit\_user\_group(array)
* מחיקת משתמש קבוצה delete\_user\_group()
* מחיקת פעילות deleteActivity(id)
* מחיקת פרויקט delete\_project(id)
* 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**







# 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 הישראל ל-PHP**

\* **UDEMI**

\* **3wSchools**

# 8. נספחים

## 8.1 הוראות התקנה

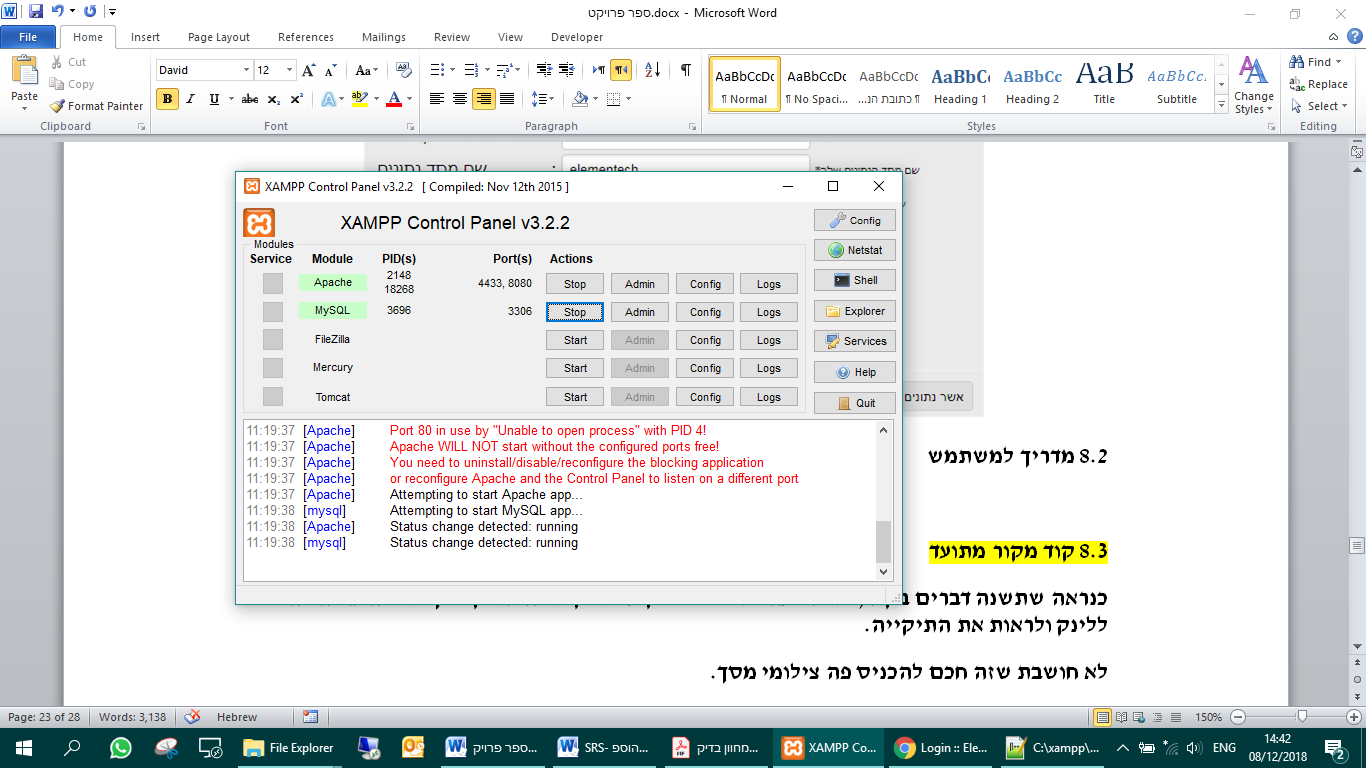
## 8.2 מדריך למשתמש

הוראות להתקנת המערכת:

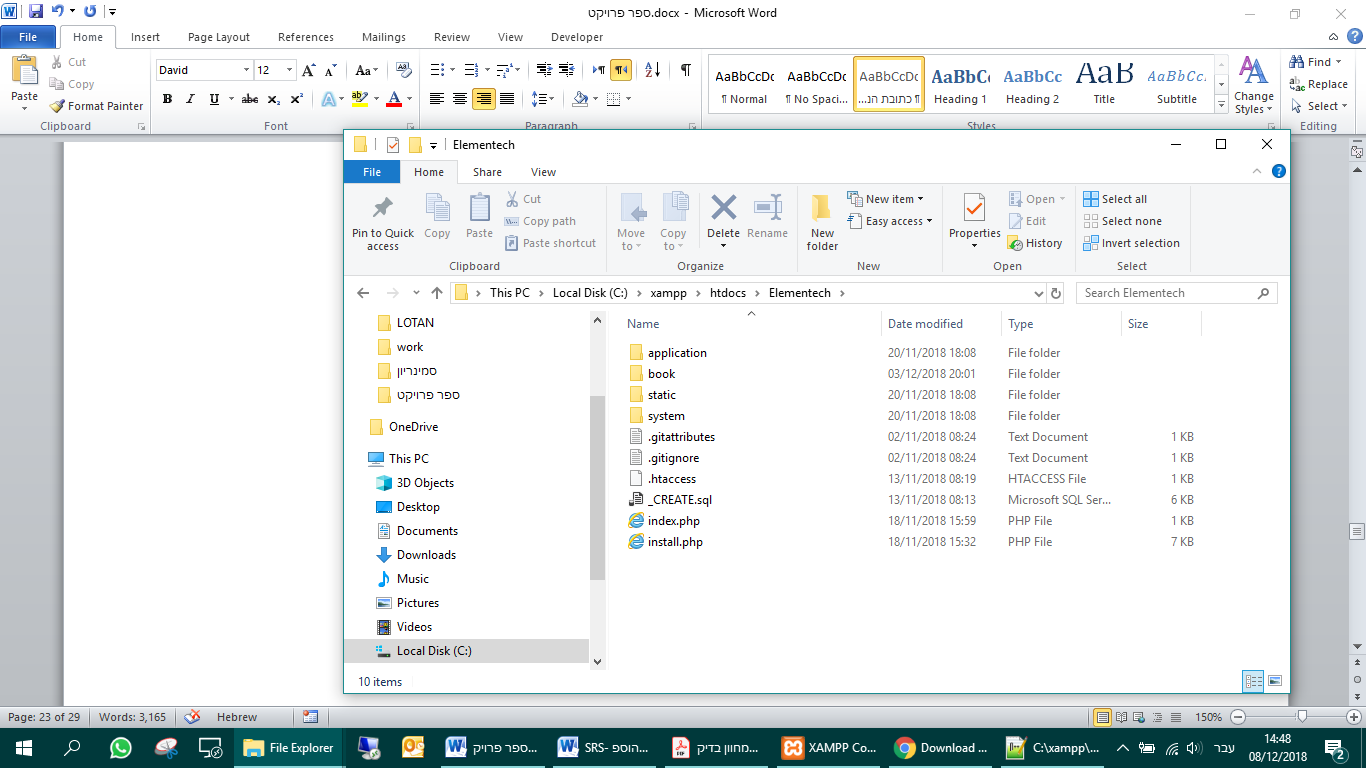
* יש להוריד ולהתקין את תוכנת XAMPP, ניתן להיעזר בלינק הבא:

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

* יש להדליק את ה-service, Apache ו-MySQL.

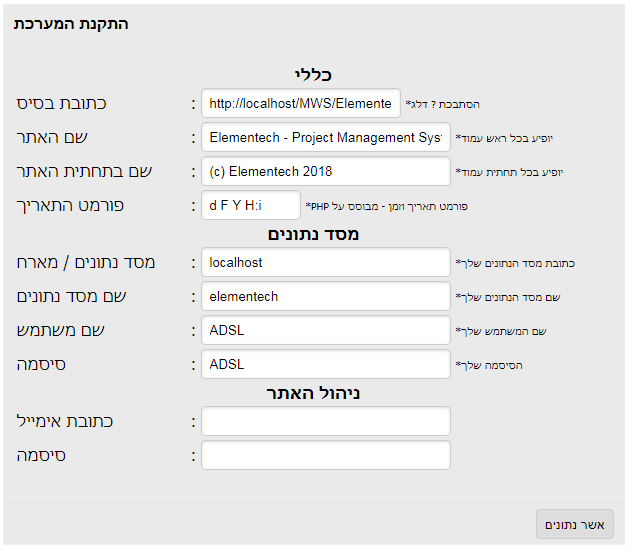


* לאחר מכן יש לגרור את תיקיית הפרויקט Elementech לנתיב הבא: C:\xampp\htdocs

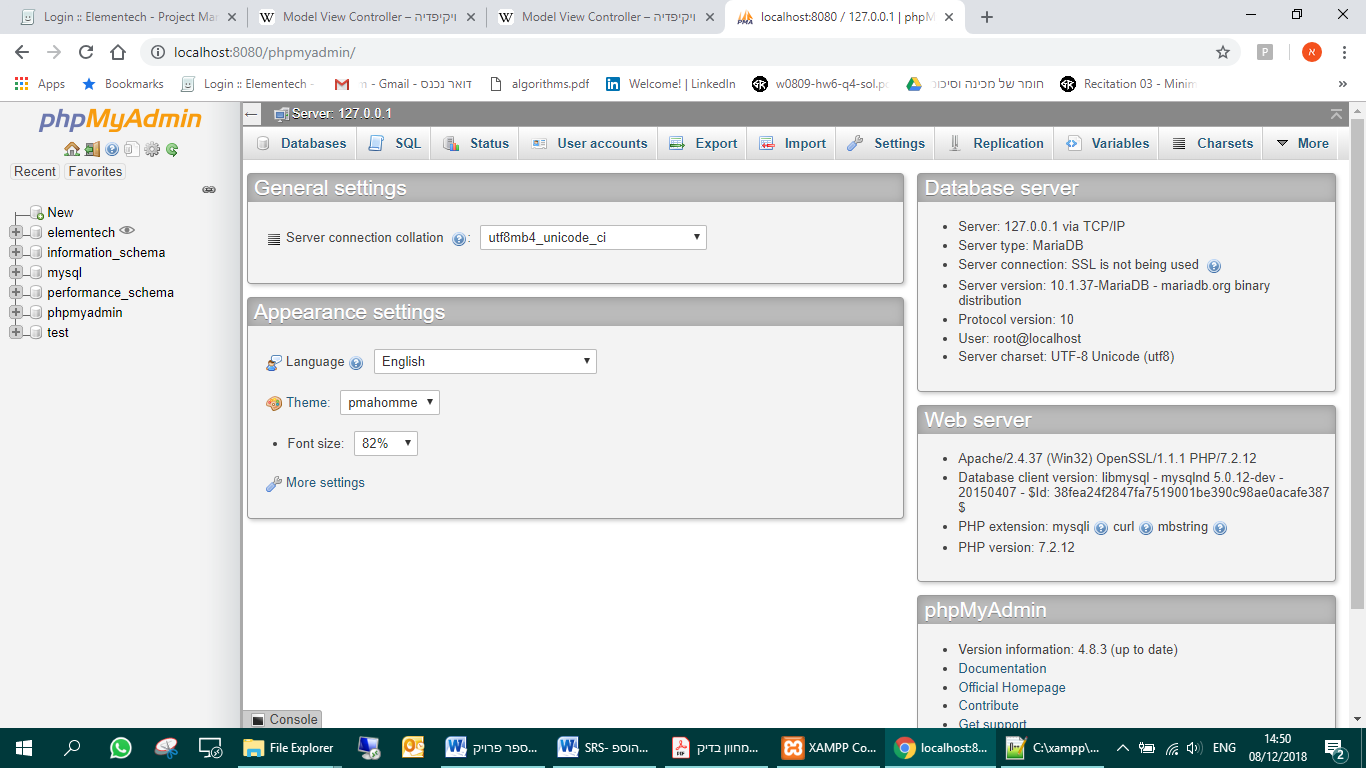


* גש לנתיב: <http://localhost/elementech/install.php> ועקוב אחר ההוראות.

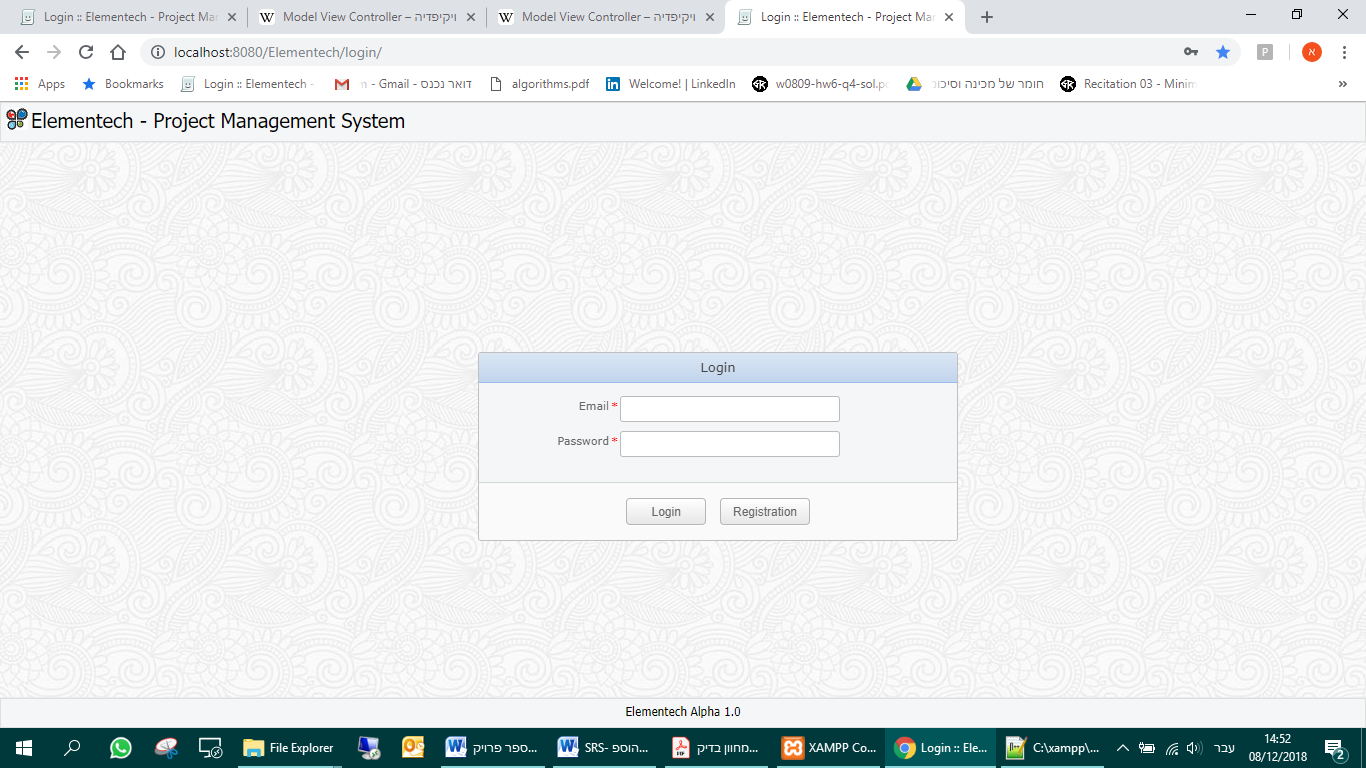
מלא את כל השדות הדרושים.



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



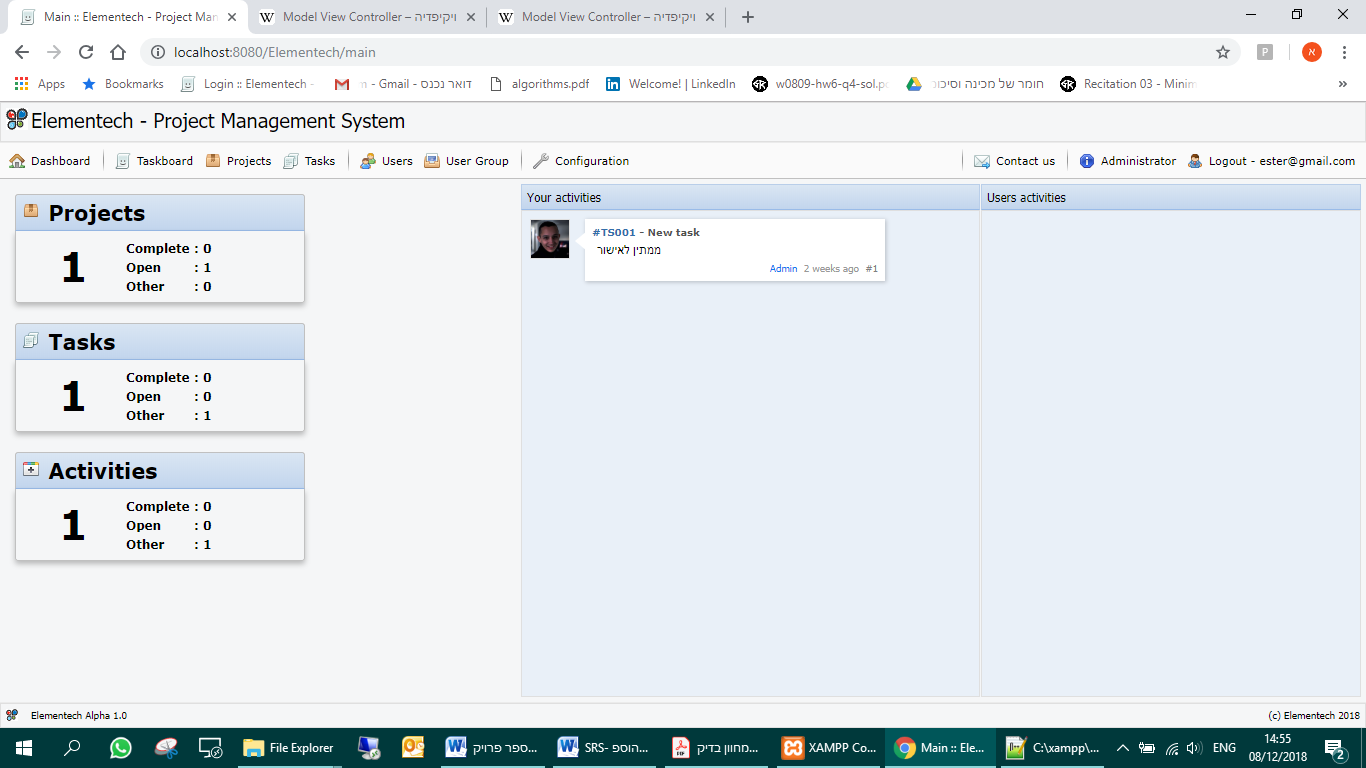
* כדי להיכנס למערכת ניהול הפרויקטים יש להכניס בשורת ה-URL את הכתובת הבאה:
* [http://localhost/MWS/Elementech/](http://localhost/MWS/Elementech/project)



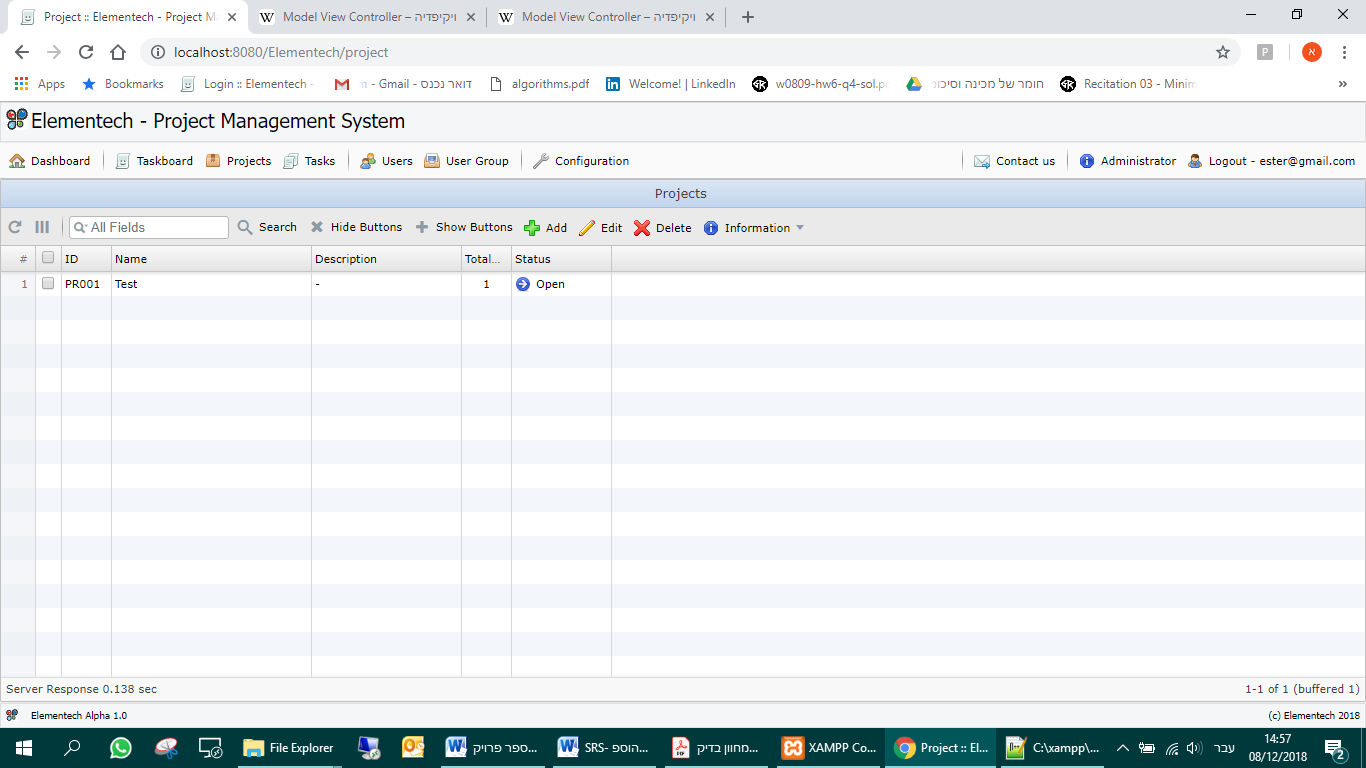
אימייל: [ari.perez25.06@gmail.com](mailto:ari.perez25.06@gmail.com) – או מה שקבע המנהל בהתקנת המערכת.

סיסמא: 250691 – או מה שקבע המנהל בהתקנת המערכת.

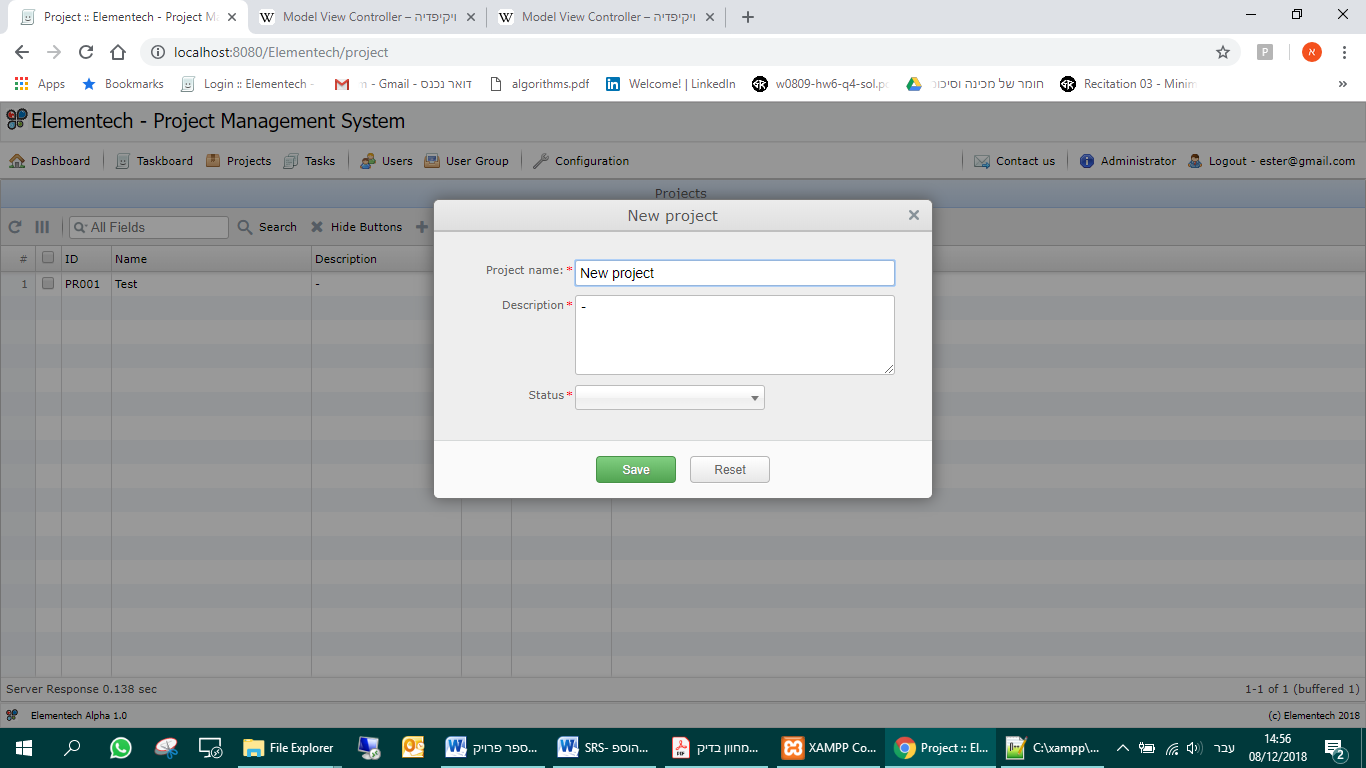
* לאחר מכן תועבר לדף אינדקס הבא:



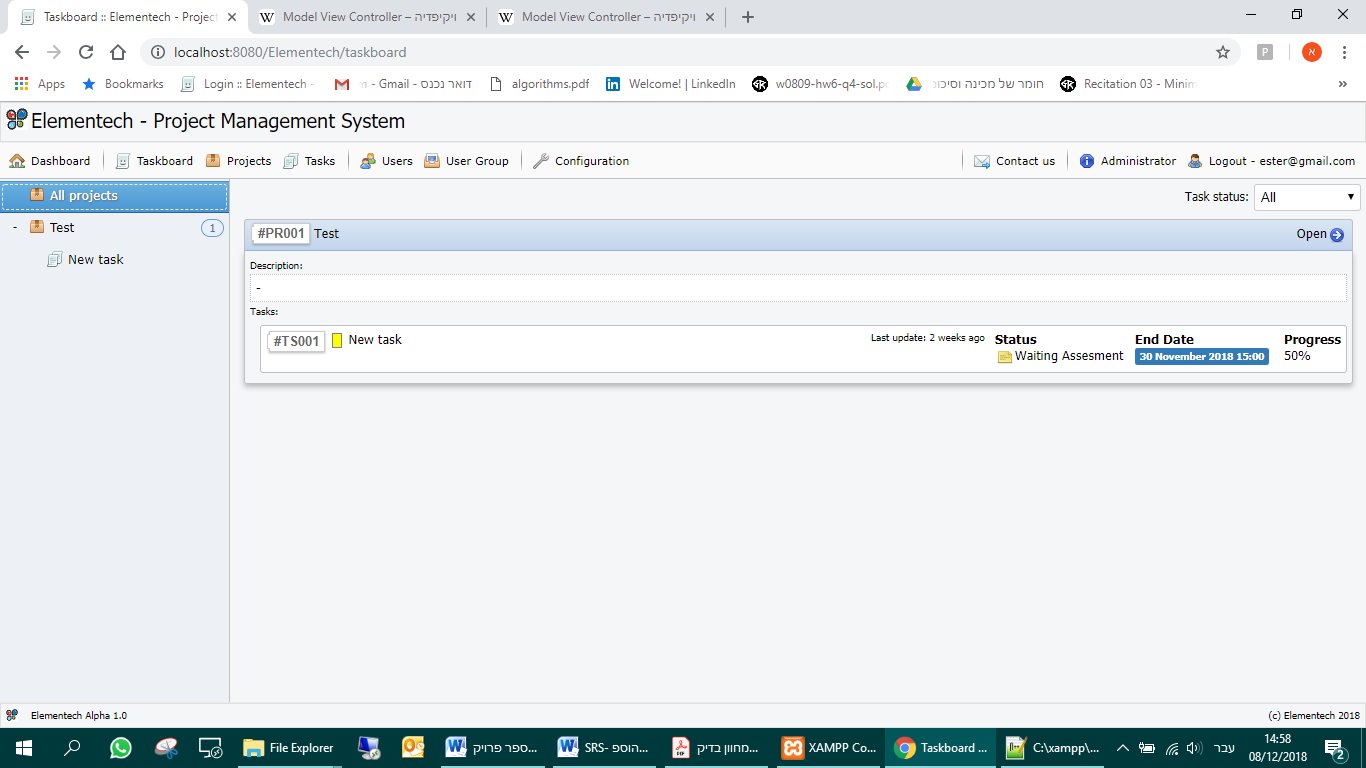
* כדי ליצור פרויקט חדש יש לגשת למסך Projects וללחוץ על המקש Add המופיע בירוק על המסך.

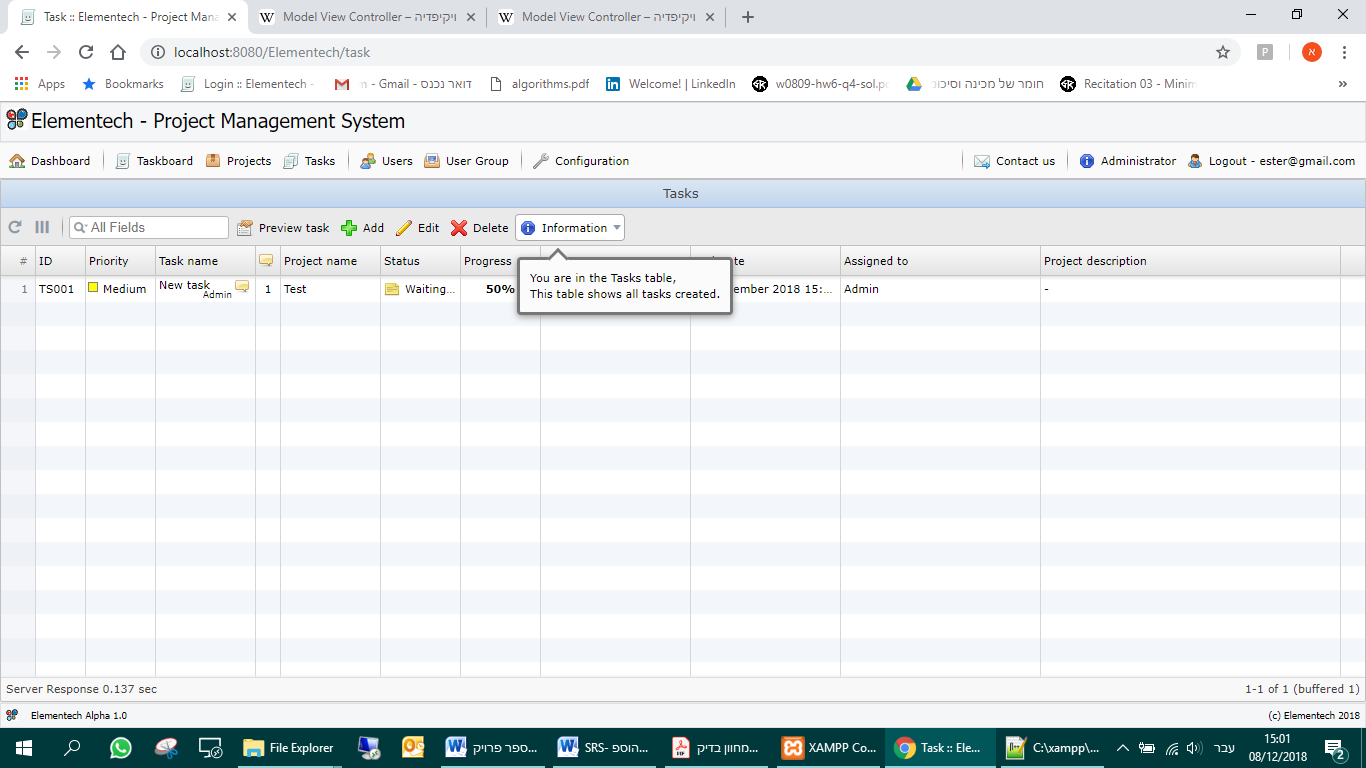


* לאחר לחיצה על כפתור ההוספה ייפתח חלון חדש שבו המשתמש יוכל להכניס פרטי פרויקט ולשמור.



* 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 לפונקציונאליות 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-phpMyAdmin.

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 מבססות 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 24\7 מכל מכשיר עם גישה לאינטרנט.
* יש לבצע גיבויים אוטומטים 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 שינתנו 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\משימות\פעילויות עבור משתמש.
* על המערכת לספק גישה למידע רלוונטי, dashboard פרויקטים\משימות\משתמשים.
* על המערכת לתת אפשרות של רישום משתמשים חדשים.
* על המערכת לתת למשתמש אפשרות להודיע על באג.
* על המערכת לתת למנהל אפשרויות ניהול משתמש\פרויקט\משימה\פעילות.
* 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](https://he.wikipedia.org/wiki/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) [שרת-לקוח](https://he.wikipedia.org/wiki/%D7%A9%D7%A8%D7%AA-%D7%9C%D7%A7%D7%95%D7%97) שבה ההצגה, העיבוד, וניהול ה[נתונים](https://he.wikipedia.org/wiki/%D7%A0%D7%AA%D7%95%D7%A0%D7%99%D7%9D) DB, הם תהליכים נפרדים מבחינה לוגית. השימוש הנפוץ ביותר בארכיטקטורה רב-שכבתית היא **ארכיטקטורת שלוש שכבות.**

מודל שלוש השכבות היא ארכיטקטורת [שרת-לקוח](https://he.wikipedia.org/wiki/%D7%A9%D7%A8%D7%AA-%D7%9C%D7%A7%D7%95%D7%97) שבה [ממשק המשתמש](https://he.wikipedia.org/wiki/%D7%9E%D7%9E%D7%A9%D7%A7_%D7%9E%D7%A9%D7%AA%D7%9E%D7%A9), הלוגיקה, והגישה אל הנתונים מפותחים ומתוחזקים כ[מודולים](https://he.wikipedia.org/wiki/%D7%9E%D7%95%D7%93%D7%95%D7%9C_(%D7%AA%D7%95%D7%9B%D7%A0%D7%94)) נפרדים, בדרך כלל על גבי [פלטפורמות](https://he.wikipedia.org/wiki/%D7%A4%D7%9C%D7%98%D7%A4%D7%95%D7%A8%D7%9E%D7%AA_%D7%9E%D7%97%D7%A9%D7%95%D7%91) שונות. מודל שלוש השכבות היא [ארכיטקטורת תוכנה](https://he.wikipedia.org/wiki/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) ו[תבנית עיצוב](https://he.wikipedia.org/wiki/%D7%AA%D7%91%D7%A0%D7%99%D7%AA_%D7%A2%D7%99%D7%A6%D7%95%D7%91).

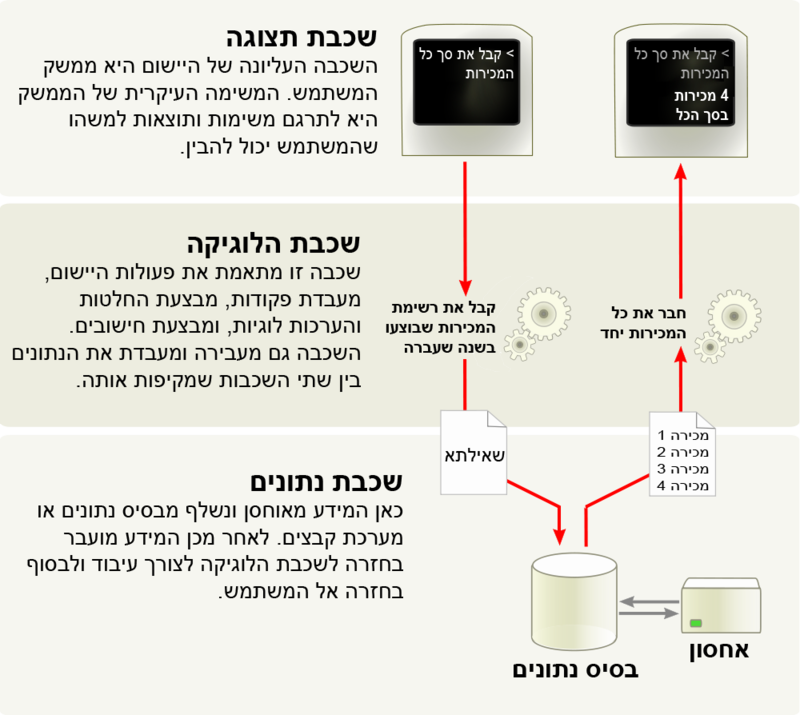
לארכיטקטורה זו ישנם יתרונות רבם ומשמעותיים, כמו תחזוקה, ניתן לבצע בדיקות בכל אחת מהשכבות בצורה נוחה ופשוטה, מקנה אפשרות להגדיר חוקים כללים עבור כל שכבה, ופשוטה ונוחה לעדכון הקוד, במיוחד שמדובר שעבודה של כמה מפתחים.

מלבד היתרונות הרגילים, ארכיטקטורת שלוש השכבות מאפשרת גם החלפה או שדרוג של כל אחת מהשכבות באופן בלתי תלוי בשכבות האחרות, בהתאם לדרישות או לשינויים ב[טכנולוגיה](https://he.wikipedia.org/wiki/%D7%98%D7%9B%D7%A0%D7%95%D7%9C%D7%95%D7%92%D7%99%D7%94). לדוגמה, החלפת [מערכת ההפעלה](https://he.wikipedia.org/wiki/%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%94%D7%A4%D7%A2%D7%9C%D7%94) בשכבת התצוגה תשפיע רק על ה[קוד](https://he.wikipedia.org/wiki/%D7%A7%D7%95%D7%93_%D7%9E%D7%A7%D7%95%D7%A8) 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](https://he.wikipedia.org/wiki/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) שהיחידות הבסיסיות שלה הן אירועים. השימוש בארכיטקטורה זו הוא במקרים רבים בשילוב עם [ארכיטקטורה מוכוונת שירותים](https://he.wikipedia.org/wiki/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%94_%D7%9E%D7%95%D7%9B%D7%95%D7%95%D7%A0%D7%AA_%D7%A9%D7%99%D7%A8%D7%95%D7%AA%D7%99%D7%9D)  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](https://he.wikipedia.org/wiki/%D7%AA%D7%95%D7%9B%D7%A0%D7%99%D7%AA_%D7%9E%D7%97%D7%A9%D7%91) 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, ונותנות 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 המשמש 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 ב[שפת](https://he.wikipedia.org/wiki/%D7%A9%D7%A4%D7%AA_%D7%AA%D7%9B%D7%A0%D7%95%D7%AA) [PHP](https://he.wikipedia.org/wiki/PHP) על בסיס [מסד נתונים](https://he.wikipedia.org/wiki/%D7%9E%D7%A1%D7%93_%D7%A0%D7%AA%D7%95%D7%A0%D7%99%D7%9D) 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](https://he.wikipedia.org/wiki/%D7%A7%D7%99%D7%93%D7%95%D7%9D_%D7%90%D7%AA%D7%A8%D7%99%D7%9D_%D7%91%D7%9E%D7%A0%D7%95%D7%A2%D7%99_%D7%97%D7%99%D7%A4%D7%95%D7%A9) ל[מנועי חיפוש](https://he.wikipedia.org/wiki/%D7%9E%D7%A0%D7%95%D7%A2_%D7%97%D7%99%D7%A4%D7%95%D7%A9), מאפשרת מעקב אחרי שינויי עריכה, מכילה מערכת תגובות מובנית, מערכת לניהול קובצי מדיה ועוד. כמו כן ארכיטקטורת WORDPRESS תומכת בשפות רבות.

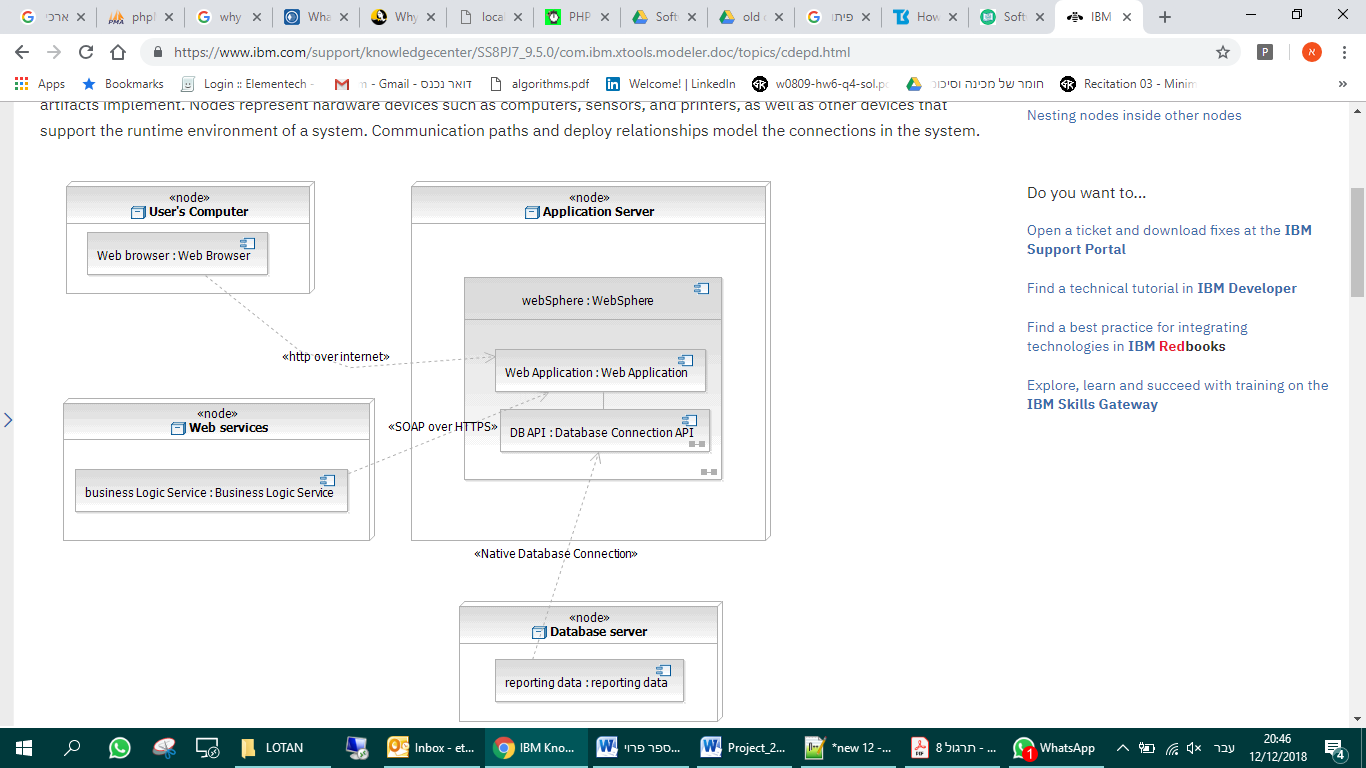
**WIX code -** מערכת שבאמצעותה ניתן לפתח בסיס נתונים דינמי, וכך להוסיף לאתר שבנית ב-WIX אין סוף תכנים שיוצגו לפי דרישת הגולש ולא צריך לבנות אותם עצמאית דף אחר דף. כלומר WIX פיתחו מערכת שמאפשרת שליפת נתונים מבסיס נתונים ובנייה שלהם בצורה אוטונומית בתוך דף אתר. האלמנט הזה חוסך זמן רב ומאפשר למשל לבנות 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. כמובן שיש צורך בתמיכה של PHP בשרת.

# 10. טופולוגית 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. ארכיטקטורה נבחרת

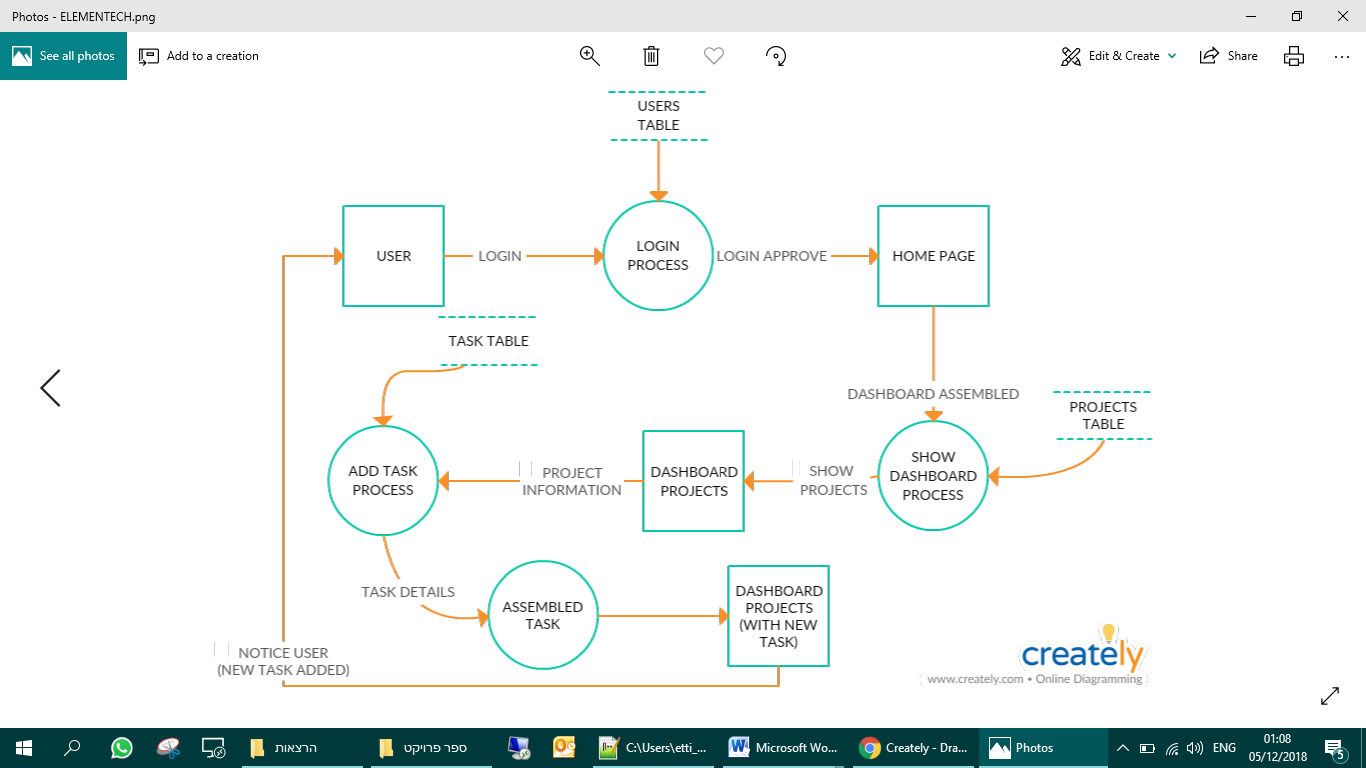
ארכיטקטורה EDA, מוכוונת אירועים היא סגנון ארכיטקטוני המגדיר [ארכיטקטורת תוכנה](https://he.wikipedia.org/wiki/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) שהיחידות הבסיסיות שלה הן אירועים. השימוש בארכיטקטורה זו הוא במקרים רבים בשילוב עם [ארכיטקטורה מוכוונת שירותים](https://he.wikipedia.org/wiki/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%94_%D7%9E%D7%95%D7%9B%D7%95%D7%95%D7%A0%D7%AA_%D7%A9%D7%99%D7%A8%D7%95%D7%AA%D7%99%D7%9D)  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

### 11.1.1 Data Flow

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, והיוותה 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-java script כדי שנוכל להשתמש בממשק המשתמש שרצינו W2UI, 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 מציגה 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 STP-DOC

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](https://he.wikipedia.org/wiki/%D7%A7%D7%95%D7%93_%D7%A4%D7%AA%D7%95%D7%97) ו[חופשי](https://he.wikipedia.org/wiki/%D7%AA%D7%95%D7%9B%D7%A0%D7%94_%D7%97%D7%95%D7%A4%D7%A9%D7%99%D7%AA) ל[בדיקות יחידה](https://he.wikipedia.org/wiki/%D7%91%D7%93%D7%99%D7%A7%D7%95%D7%AA_%D7%99%D7%97%D7%99%D7%93%D7%94) בשפת PHP. אם במידה ואחד מהצוות נתקל בשגיאה, עליו לתקן את השגיאה במיידי, 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 הפונקציות האחרות להצגת 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 |