# שם הפרויקט :

עוד לא היה ברית

# רקע

## **תיאור ורקע כללי**

מערכת מקוונת המיועדת בעיקר לרשת פנים ארגונית, ולקהל יעד ממוקד.

היא נועדה לשמש ככלי לניהול החזרת אבדות לבעליהם, באופן נגיש ומסודר.

המערכת דינאמית ומאפשרת התאמות פנים ארגוניות על פי צרכיו של כל לקוח אופציונלי.

## **מטרות המערכת**

## לספק כלי לפרסום מידי של מציאות במוסד לימודי או בכל מוסד ארגוני אחר, במטרה להקל על ניהולו השוטף של המוסד.

## ליצור מערכת אמינה זמינה ופשוטה לשימוש שתאפשר מציאה לפריטים אישיים של יחידים במוסד ותהווה כתובת בלעדית לפרסום כל מציאה.

# סקירת מצב קיים בשוק

כיום, מטפלים ארגונים ומוסדות רבים באבידות בדרך של פרסומים ספציפיים במקומות מרכזיים, או שלא מפרסמים כלל. במוסדות רבים אין ממש למי לפנות כדי להעזר במציאת ובפרסום אבדות,  
מערכת האבדות מהווה במוסדות רבים מטרד לא קטן, הן לצוות הניהול והן לבעלי אבדות, או למוצאים.  
הנהלת המוסד נדרשת לטפל בעשרות אם לא במאות ויותר מזה חפצים שאין לדעת אם ימצא להם דורש מתי והיכן, כל ההתנהלות מורכבת, לא מסודרת ויוצרת מצב לא אידיאלי בעליל.  
כשחבר במוסד מוצא חפץ יקר ככל שיהיה, אין לו דרך מסודרת ואחראית לידע את **כל** חברי המוסד האחרים במציאה במטרה למצוא את המאבד,  
קל וחומר שכשאדם מבין שאבד לו דבר מה, אין לו אפילו קצה חוט למציאת אבדתו, היא עשויה להתגלגל מיד ליד במשך זמן רב וקלוש הסיכוי למציאתו.

# מה הפרויקט אמור לחדש או לשפר

הפרויקט מציע מערכת לניהול אבדות ומציאות, ידידותית ונגישה, המאפשרת פרסום של כלל המציאות במוסד, דפדוף מהיר וחיפוש יעיל שינגיש למאבד את אבדתו ויחסוך עשרות העברות ידיים וחיפושים עצמאיים מורכבים.

המערכת תהיה נגישה לכל חבר במוסד ותהיה גמישה להתאמות פנים ארגוניות ע"י הנהלת הארגון.

מערכת מקוונת זו עתידה להעלות משמעותית את אחוזי מציאת האבדות, הודות למאגר האחיד המוחלט והידיעה הברורה שכשמשהו אבד, המקום הטבעי שלו להיות – זה במאגר הזה.

כמו כן המערכת תקל משמעותית על ניהול מוסדות בכך שתסיר מעליהם את עול הטיפול באבדות בצורה מוחלטת.

# דרישות מערכת ופונקציונאליות

## **דרישות מערכת**

## המערכת תושתת על רשת איטריום ותדרוש התקנת ארנק MetaMask (תוסף על הדפדפן) – לצורך התחברות לרשת הבלוקצ'יין של איטריום. כמו"כ יעשה שימוש בחיישן טביעת אצבע (כיום מורכב על מחשבים ניידים, ועל טלפונים חכמים ) ותיערך התממשקות ל API של משרד הפנים לצורך זיהוי טביעת אצבע האזרח ואימות זכות הצבעתו.

## **פונקציונאליות**

מנהל:

* + בוחר סיסמת מנהל
  + מזין כתובת מייל רשמית למוסד
  + מזין רשימת מיקומים על פי חלוקה מסודרת של המוסד

מוצא:

* + מזין פרטי אבידה מלאים כולל בחירת קטגוריה.
  + מזין פרטים אישיים לפניה ישירה של המאבד.
  + מעדכן סטטוס מציאה ל"הוחזרה לבעליה" כשנמצא המאבד.

# בעיות צפויות במהלך הפיתוח

## הבעיות:

### בעיה 1: טכנולוגיית הבלוקצ’יין כיום נאבקת עם סוגיית ה Scalibility , כלומר כמות התעבורה והטרנזאקציות שיכולות לזרום על גבי בלוקים בזמן נתון עדיין מוגבלת ולא יכולה לשרת מסה של משתמשים.

### בעיה 2: תהליך ההצבעה כרוך בסריקת טביעת אצבע, דבר שלא זמין כיום לכלל האוכלוסיה.

## פתרונות אפשריים:

### לבעיה 1:

### פתרון 1: כנכתב לעיל, הפרויקט הינו **Proof of Concept** ואינו מתימר לפתור בעיה זו.

* + פתרון 2 : יצירת חוות שרתים מבוזרת בגודל הרצוי שיאחסנו את המידע,השרתים יתקשרו על אותו עיקרון שבלוקצ'יין מביא עימו (שלכל שרת יש את שרשרת הבלוקים של המידע, וכל בלוק מכיל את ערך הגיבוב של הבלוק לפניו).

### לבעיה 2:

### פתרון 1: הצבת עמדות רחוב עם אפשרות גישה למערכת.

### פתרון 2: אפשרות הזדהות עם מספר תעודת זהות.

### פתרון 3: עד שייושם רעיון הבלוקצ'יין גם טכנולוגיה זו עתידה להתפתח ולהפוך רווחת בקרב המשתמשים.

## הפתרון הנבחר עבור כל אחת מהבעיות:

### לבעיה 1: פתרון 1, הפרויקט מהווה סימולציה ומדמה רעיון , הוא נכתב בתהליכים מתפתחים המתמזגים עם התקדמות רשת הבלוקצ'יין. אנו תקווה שעד מימוש הרעיון והפיכתו למציאות- בעיה זו תיפתר כאיננה.

### לבעיה 2: פתרון 3, מאותה סיבה שנכתבה לעיל.

# פתרון טכנולוגי נבחר

## **טופולוגית הפתרון**

## המערכת תושתת כמו שנכתב לעיל על פלטפורמת הבלוקצ'יין של איטריום, תשתמש בTruffle framework ותכתב בסביבת העבודה Visual Studio Code. יעשה שימוש בסימולטור גנאש, הוא יהווה הדמיה לרשת הבלוקצ'יין המקורית.

**מהו בלוקצ'יין?**

טכנולוגית הבלוקצ'יין מוגדרת כמערכת דיגיטליות מבוזרת הפועלת כרשומה פתוחה, משותפת ואמינה של עסקאות בין גורמים, שאינה מאוחסנת בידי רשות מרכזית. לכל המשתמשים עותק והם יכולים לראות באופן שקוף ובכל רגע נתון את העברת הנתונים הנעשות במערכת. השקיפות שהטכנולוגיה מאפשרת, לצד הבקרה, העקביות ואבטחת הנתונים המוגברת יכולה לסייע בהגדלת האמון בין הצדדים, להקל את המסחר ולשפר את היעילות של העסקאות.

בלוקצ'יין מיועד להגיע להסכמה בין גורמים שונים ובלתי תלויים על בלוקים של מידע שמחוברים בשרשרת (Chain). כל בלוק כולל מזהה וגיבוב של תוכן הבלוק הקודם וכך מתחבר אליו. כך נוצר מבנה נתונים מבוזר שעל תוכנו יכולים להסכים כל הגורמים המעורבים. אין אפשרות לשנות את תוכן הבלוקים מרגע שהם נקבעו.

 אנשים שונים במחשבים שונים בונים מעין סולם דיגיטלי של טרנזקציות, הכולל נתונים מוצפנים. אחת ליחידת זמן נכתבים הנתונים לבלוק. בתום זמן מסוים, ננעל הבלוק, ומצטרף כמעין שרשרת לבלוקים הקודמים.

[קישור לערך בלוקצ'יין בויקיפדיה](https://he.wikipedia.org/wiki/%D7%91%D7%9C%D7%95%D7%A7%D7%A6%27%D7%99%D7%99%D7%9F)

ככל יישומי בלוקצ'יין, המערכת בנויה כ DAPP – Decenteralize Application, או בשם אחר - חוזה חכם, כאשר הכוונה ליישום מבוזר המתנער מהצורך של אמון בין אישי באופן מושלם.

**מהו חוזה חכם?**

חוזה חכם הוא פרוטוקול מחשב המיועד לאפשר, לאמת או לאכוף באופן דיגיטלי את המשא ומתן או ביצוע החוזה. חוזים חכמים מאפשרים ביצוע עסקאות אמינות **ללא תלות בצד שלישי**. עסקאות אלו ניתנות למעקב ובלתי הפיכות.

דיאגרמה

## **טכנולוגיות בשימוש:**

במערכת זו כתיבת **צד השרת** מתבססת על טכנולוגיית בלוקצ'יין שהיא תפיסה שונה בעולם השרתים, בכך ששמירת הנתונים אינה נעשית בשרת אחד כמקובל, אלא בקבוצת שרתים מבוזרים המתקשרים ביניהם, כל הפעולות נעשות בטרנזקציות המחיבות הסכמה של כלל המחשבים שיוצרים את הרשת.  
הטכולוגיה מתבססת על כתיבת חוזים חכמים כמו שהוסבר לעיל.

**צד הלקוח** יכתב בשפת JavaScript בשימוש ספרית React ובספרית Web3 המאפשרת אינטראקציה בין צד הלקוח לרשת איטריום.

## **שפות הפיתוח:**

### בצד השרת: שפת Solidity של איטריום

### בצד הלקוח: שפות javaScript, html5, css3

## **תיאור הארכיטקטורה הנבחרת**

השימוש בטכנולוגית הבלוקצ'יין מבטיח את אמינות הנתונים, ומונע באופן מושלם זיופים וחבלות.

שפתSolidity מאפשרת כתיבת חוזים חכמים בשפה נוחה וקריאה לאדם.

השימוש בספרית React.js המאפשרת פיתוח אפליקציות מתקדמות,והיא טכנולוגיה פופולרית ואהובה בשוק.

ספרית Web3.js היא הספריה הנדרשת לצורך האיטראקציה בין צד הלקוח לרשת איטריום.

## **חלוקה לתכניות ומודולים:**

### **בצד השרת:**

לאיטריום ישנו EVM (Etherium Virtual Machine) - זוהי מכונה וירטואלית אשר מקבלת כקלט את קוד הSolidity , מקמפלת אותו ומפיצה ל Ethereum Networ.

כל החוזים אשר רצים על גבי Ethereum Network מבודדים לחלוטים מהעולם החיצון.

החוזים החכמים מממשים את הפונקציונאליות של המערכת וכל טרנזקציה מריצה את החוזה החכם ונשמרת בשרשרת הבלוקים.

### **בצד הלקוח:**

חלוקה לוגית וארגונית לקומפוננטות כמקובל בספרית React ,

כמו כן, ספרית React עובדת על Virtual DOM , הדום הוירטואלי של React הוא שכל קוד ה HTML נכתב מתוך הJavascript , אין יותר הפרדה בין HTML ל CSS הכל משתלב לקובץ Javascript אחד או לרבים כמותו.

# סביבת השרת

## **ממשק המשתמש** יתקבל כ - HTML ויהיה מוצג באמצעות תוסף לדפדפן אינטרנט

(MetaMask)

## **ממשקים למערכות אחרות:**

## API משרד הפנים – לצורך שליפת נתוני האזרח.

## שימוש בהתקן לסריקת טביעת אצבע לזיהוי מהימן של האזרח .

## **שימוש בחבילות תוכנה:**

שימוש בספרית React.

שימוש בספרית Drizzle של איטריום.

שימוש בספרית Web3 של איטריום.

# מבני נתונים וארגון קבצים

## **שיטת האחסון**

הנתונים נשמרים על רשת הבלוקצ'יין של איטריום (במערכת הנוכחית הרשת היא הסימולטור גנאש).

## **מבני הנתונים**

הקונספט של טבלאות לא קיים, הנתונים נשמרים באוביקטים של סולידיטי לצורך ההפשטה נתאר את המודולים באופן טבלאי, למרות שלמעשה בלוקצ'יין לא תומך במבנה נתונים, כי אם באוביקטים מסוג Json:

* טבלת מפלגות והסכמי עודפים
* טבלת ישובים
* טבלת הצבעות ע"פ ישוב ומפלגה
* טבלת מצביעים (hash table)

## **מנגנוני התאוששות מנפילה/ קריסה/ תמיכה בטראנזקציות**

הרעיון של בלוקצ'יין הוא למנוע סכנות מסוג זה בעצם הוויתו, מתבסס על פונקציונאליות זהירה ביותר, והנתונים מוכפלים לכמות נרחבת של Lids

# תרשימי מערכת מרכזיים

## Use Case





## Sequence diagram





# תיאור המרכיב האלגוריתמי – חישובי

## **חלוקת המושבים בכנסת בין הרשימות המתמודדות**

## חישוב אחוז החסימה מתוך סך הקולות.

## חישוב המודד – שווי מושב בודד בכנסת (מספר הקולות למנדט).

## קביעת מספר המנדטים השלמים לכל מפלגה.

## חחלוקת הקולות העודפים למנדטים על פי נתוני ההסכמים.

# תיאור/התייחסות לנושאי אבטחת מידע

מערכת הבלוקצ'יין מאובטחת ומטפלת בשמירת הנתונים בצורה המהימנת ביותר.

# משאבים הנדרשים לפרויקט:

## מספר שעות המוקדש לפרויקט: 720

## ציוד נדרש: מחשב, חיבור לאינטרנט, סורק טביעת אצבע.

## תוכנות נדרשות: visual Studio Code, Solidity and nodeJS packeges

## ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט:

לימוד תיאורטי על מגמת הבחירות ועקרונותיה.

## לימוד מכליל על מערכת הבלוקצ'יין ומרכיביה - הלוגיקה, הטכנולוגיה, הטכניקה,

## ההתממשקות אליה, השפות הנפוצות בה והקנית שליטה מלאה בכל כליה.

## הכרות עם ספרית React מהבסיס ועד לתחושת ביטחון ותוצאות מספקות.

## ספרות ומקורות מידע:

## ללימוד :**React** ReactDocs & Scrimba

gitHub להורדת ספריות שונות לדוגמא: String.sol, w3schools.com

## להבנת טכנולוגית **הבלוקצ'יין**: חיפושי רשת וסרטוני וידאו שונים, TruffleSuite, Solidity docs & Truffle docs, DappUniversity Vidoes, blockchaintechnologies.com, investopedia.com, tutorialspoint.com, guru99.com

## לחשיפת התיעוד הפונקציונאלי של**Web3** : Read The Docs

# תכנית עבודה ושלבים למימוש הפרויקט:

|  |  |  |
| --- | --- | --- |
| **שלב** | **משך זמן לעבודה זוגית** | **תאריכים משוערים** |
| * + ייזום הרעיון | **30 שעות** | **1.6-5.6** |
| * + ניתוח מערכת | **50 שעות** | **6.6-12.6** |
| * + ניתוח מבנה נתונים | **20 שעות** | **13.6-18.6** |
| * + אפיון UX - UI | **7 שעות** | **18.6-20.6** |
| * + כתיבת הלוגיקה העסקית | **150 שעות** | **21.6-10.10** |
| * + כתיבת ממשק המשתמש | **40 שעות** | **21.6-10.10** |
| * + עיצוב | **40 שעות** | **21.6-10.10** |

**חתימת הסטודנט חתימת המנחה האישי**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

הערות ראש המגמה במכללה:

אישור ראש המגמה במכללה:

שם:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ תאריך:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

הערות הגורם המקצועי מטעם מה"ט:

אישור הגורם המקצועי מטעם מה"ט:

שם:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ תאריך:\_\_\_\_\_\_\_\_\_\_\_\_\_\_