

**עבודת גמר 5 יח"ל**

**נושא העבודה :** מערכת מבוססת IOT לניטור וטיפול בגינון ביתי

**שם תלמיד :** דורון מאור

**ת.ז תלמיד :** 215163403

**שם בית ספר ועיר :** קריית החינוך ע"ש עמוס דה-שליט, רחובות

**שם המנחה :** ערן בינט

**מועד הגשה :** 24.12.2022

תוכן עניינים

[1. מבוא..................................................................................................................................... 3](#_Toc122374461)

[2. תיאוריה.......................................................................................................................................... 4](#_Toc122374462)

[3. תוצר סופי 6](#_Toc122374463)

[4. תהליך כתיבת הפרויקט 12](#_Toc122374464)

[5. מרכיבי פתרון 12](#_Toc122374465)

[6. תסריטי בדיקה 14](#_Toc122374466)

[7. רפלקציה 14](#_Toc122374467)

[8. הוראות התקנה ותפעול 15](#_Toc122374468)

[9. ביבליוגרפיה 15](#_Toc122374469)

[10. נספחים 16](#_Toc122374470)

1. מבוא
   1. נושא העבודה

נושא הפרויקט הוא מערכת מבוססת IOT (Internet of things) לניטור וטיפול מרחוק בגינון ביתי. המערכת מאפשרת לבעלי הבית לבקר על גדילת הצמחייה הביתית (עציצים ושתילים). למערכת יכולות טיפול החורגות מיכולת טיפול הצמח על ידי בן אדם. המערכת תכלול בקרת תאורה ולחות, השקיה והתאמת תאורה על פי מצב הצמח. ההחלטות הטיפוליות על הצמח יתבצעו על ידי אלגוריתם מתפתח השואף לספק לצמח את צרכיו האופטימליים על פי סוגו.

המערכת עושה שימוש נרחב בבקר Arduino ורכיבי פלט/קלט המאפשרים את הטיפול בצמח, ושמה דגש על הפקת דוחות מתקדמים המשקפים את גדילת הצמח והטיפול בו. כמו כן, הדוחות כוללים מידע הקשור לפעולות המערכת אשר מתבצעות מרחוק על כל צמח.

שמה של המערכת הוא MyGardenGenie (בתרגום ישיר: הג'יני של גינתי). השם ההומוריסטי מבטא את היכולות הקסומות של המערכת בנושא טיפול הצמח.

* 1. מטרות מרכזיות

המטרות המרכזיות של הפרויקט:

* זיהוי הצמח – המערכת תזהה את הצמח ותתאים עבורו תנאי לחות ואור על מנת לאפשר טיפול אופטימלי בצמח ככל שניתן.
* מתן טיפול לצמח – מתבצע על ידי השקיה ותאורה מתאימים על פי נתונים שנאספים על ידי חיישני לחות ותאורה המשוקללים בצרוף לתנאי גידול הצמח המיטביים.
* יישום מערכת לומדת המתבססת על מערך נתונים בשרת (הכולל מידע ממשתמשים רבים על גדילת הצמח בתנאים שונים), הנבנה לצורך שיפור אלגוריתם טיפול הצמח.
* ממשק משתמש ידידותי ומאובטח – צפייה בדוחות אודות הצמח, ביצוע פעולות טיפוליות שונות על הצמח, תוך מניעת גישה למשתמשים לא מורשים.

מטרותיי האישיות מן הפרויקט:

* התמקצעות בתכנות מול בקר Arduino – העברת החומר הנלמד במהלך הקורס אל הממד החומרתי. ופיתוח יכולותיי בתחום זה לרמה מקצועית לטובת הפרויקט הנוכחי ופרויקטים עתידיים.
* הגדרת ובניית הפרויקט - המוצר יתבסס על התכנון המשתקף בספר הפרויקט, באופן שממצה את כישוריי ויכולותיי כתלמיד מגמת סייבר.
* רמת יישום הפרויקט - יישום פרויקט ברמה של מוצר מדף הנרכש כמערכת שלמה.
* לוחות זמנים – עמידה בלוחות זמנים ואבני הדרך של הפרויקט.
  1. רציונל

לבניית ופיתוח הפרויקט שלי, קיימות מספר מוטיבציות אשר הניעו אותי לפתח את הפרויקט הנ"ל.

ראשית, מטרתי העיקרית ליצירת הפרויקט היא הדרישה בשוק למוצר שכזה. אחת מתופעות הלוואי של משבר הקורונה והסגרים הביאו עמם זינוק של עשרות אחוזים במכירות של צמחים ועציצים לגידול ביתי ובמיוחד עבור צעירים. לאחר שהוסרו הגבלות הקורונה, חזרו האנשים לבלות מחוץ לבית ולשהות בחו"ל. הם נתקלו בבעיה בה כל מגדל צמחים ביתי נתקל בה – כיצד להמשיך לטפל בצמחים. הרי אי אפשר להזניח אותם ללא כל טיפול לאורך זמן. בדרך כלל, המגדלים משאירים את הצמח אצל שכנים, חברים או בני משפחה או לחלופין משאירים בידיהם מפתח לדירתם על מנת לטפל בצמחם.

שנית, ברמה האישית, בחרתי פרויקט הכולל חומרה לנוכח אהבתי לעולם זה. מאז ומתמיד התעסקתי עם בקר ה Arduino, אך מעולם לא בניתי פרויקט גדול בשימושו. לכן, בהזדמנות זו, החלטתי לבחור בפרויקט לא שגרתי המשלב את שני תחביבי הגדולים – תכנות וחומרה.

* 1. קישור לחומר הנלמד

העבודה מתייחסת לחומר הנלמד במספר רב של תחומים.

* תקשורת - רמה 4 במודל 7 השכבות – במהלך כיתה יא', למדנו על תהליך התקשורת ברשת, פרוטוקולים שונים הקיימים ברשת ועל התקפות שונות בה. בפרויקט אממש תקשורת בפרוטוקול TCP/IP באמצעות sockets של לקוחות שונים מול שרת אחד מרכזי.
* ממשק משתמש - Flask – במהלך כיתה יא', למדנו על העקרונות הקשורים לבניית ממשק משתמש וכיצד ליישם אותם ב Python. בפרויקט אממש את הגרפיקה באמצעות ספריית Flask.
* בסיסי נתונים – במהלך הקורס, למדנו כיצד לתקשר מול סוגי מסדי הנתונים השונים בשפת Python. בפרויקט אממש זאת במסגרת פיתוח שרתי SQL של המשתמשים הרשומים ונתוני הצמח האופטימליים של זני צמחים מסוימים. השימוש במסד הנתונים מסוג SQL נבחר מכיוון שקיימת תמיכה מובנת של שימוש במסד SQL בספריית Flask.

דוחות הבקרה של הצמחים הפעילים יאוחסנו במסד נתונים מסוג NOSQL מכיוון שמסד נתונים זה בעל יכולות עדכון ויצירת מידע מהירים לעומת SQL.

* Threads – במהלך שנת הלימוד הקודמת, למדנו על שימוש מתקדם בספריית thread בשפת python. בפרויקט אעשה שימוש נרחב בספרייה זו.
* אבטחה – במהלך כיתה יא', למדנו על עקרונות אבטחת מידע ברשת וכיצד לאבטח שמירת נתונים בשרת. בפרויקט אממש זאת בכך שאאבטח את פקודות המשתמש ללקוח ואבצע הצפנה חד כיוונית בשמירת שמות המשתמשים וסיסמתם.

1. תיאוריה
   1. תיאוריה

בהמשך

* 1. מוצרים קיימים

כיום בשוק, קיימים מוצרים רבים אשר מטרתם העיקרית היא ביצוע השקיה אוטומטית. המערכות אשר מבוססות בעיקר על חומרה הן ללא חיבור לשרת, ללא אלגוריתם מתקדם לטובת טיפול הצמח ואף ללא הפקת דוחות למשתמש.

אחד התחליפים העיקריים למוצר שלי, הינו ה [FarmBot](https://farm.bot/).

ה FarmBot הינו מערכת חקלאות המשתמשת ברובוטיקה ואוטומציה כדי לשתול, לגדל ולקצור יבולים בקנה מידה קטן. המערכת מורכבת מפלטפורמה רובוטית הנעה לאורך מסילות ומצוידת בכלים וחיישנים לשתילה, השקיה וניכוש עשבים. היא מסוגלת לטפח מגוון גידולים, כולל ירקות, עשבי תיבול ועצי פרי קטנים. למערכת ממשק מבוסס web כדי לאפשר למשתמשים להגדיר את לוחות הזמנים הרצויים לשתילה והשקיה ולעקוב אחר התקדמות גדילת היבולים. הטכנולוגיה שבה משתמש ה FarmBot כוללת חיישנים ומנועי דיוק, המאפשרים שתילה וטיפול ראוי בגידולים. מערכת זו ניתנת להרחבה, כלומר ניתן להפעיל אותה במגוון הגדרות וניתן להתאים אותה לצרכים הספציפיים של המשתמש. המחיר של מערכת FarmBot משתנה בהתאם לדגם הספציפי ולתכונות הרצויות, אך עלות הדגם הזול ביותר של ה FarmBot הינו 1695$ (5871 ₪).

בשוק קיים מוצר נוסף, אשר דומה למוצר שלי אך בעל יכולות מצומצמות יותר - [RainPoint WiFi Automatic Watering System.](https://www.rainpointonline.com/products/ik10pw-wl-fi-controlled-water-tank-pump-timer) מערכת זו מבית Rainpoint הינה מערכת השקיה אוטומטית עבור עציצים שנועדה לעזור למשתמש לטפל בצמחיו על ידי מתן כמות מים קבועה. מערכת זו מורכבת מיחידת בקרה המחוברת לסדרה של צינורות המובילים מים לצמחים. יחידת הבקרה מתוכנתת להשקות את הצמחים לפי לוח זמנים. המוצר בעל יכולת השקיה דרך צינור יחיד היכול להתפצל למספר רב של עציצים. מערכת זו שימושית במיוחד עבור אנשים שאינם מסוגלים להשקות את הצמחים שלהם באופן קבוע עקב התחייבויות עבודה או נסיעות, או עבור אלה שיש להם מספר רב של צמחים ורוצים להבטיח שכולם מקבלים את הטיפול הראוי. למוצר אפליקציה המאפשרת לשנות את הגדרות ההשקיה וקבלת התראות. עלות המוצר היא 55$ (190 ₪).

MyGardenGenie בעלת חוזקות רבות יותר בתחום הדוחות למשתמש המעידים על גדילת הצמח, טיפול בצמח באופן מידי מרחוק ופשטות של ההתקנה. זיהוי הצמחים המובנה מקל על התקנת המערכת ותחילת הטיפול בצמח. יתר על כן, במערכת קיים אלגוריתם למידה ממוחשבת המשנה את תנאי טיפול הצמח בהתאם לנתוני השטח ממשתמשים אחרים. עלות אבטיפוס המערכת תעלה כ 100 ₪ לכמות ייצור גדולה. אחד החסרונות המשמעותיים והניכרים במערכת, הוא הצורך במחשב ומצלמה אשר יבקרו על הצמחים. היום בשוק קיימים לוחות בקר Arduino חדשניים עם מצלמה וכרטיס רשת מובנים, כך שניתן בעתיד לוותר על נוכחתם של המחשב והמצלמה. כמו כן, בניגוד ל FarmBot. המערכת בנויה לעציצי בית קטנים ולא לגינות או אדניות גדולות

ניתן לייצג את ההבדלים בטבלה:

|  |  |  |  |
| --- | --- | --- | --- |
| **שם המוצר** | **MyGardenGenie** | **RainPoint Automatic Watering System** | **FarmBot** |
| שליטה מרחוק | אפשרי, ניתן ליישם שרת בעל כתובת IP לא מקומית ובכך לשימוש מחוץ לבית | אך ורק בתוך הרשת הביתית | כן |
| אלגוריתם טיפול בצמח מתעדכן (למידה חישובית) | האלגוריתם מתעדכן באופן אוטומטי על בסיס נתוני השטח. כמו כן, האלגוריתם ניתן לעדכון על ידי המנהלים ועל ידי משתמשים רשומים | אין | האלגוריתם מתעדכן באופן אוטומטי על בסיס נתוני השטח |
| הפקת דוחות למשתמש | דוחות המציגים מידע רב על התפתחותו של הצמח ופעולות שהתבצעו | אין | דוחות המעידים על פעולות אשר המערכת ביצעה בלבד |
| אופציית שתילת וזריעת צמחים | אין | אין | עד 2 סוגי זרעים בו זמנית |
| ממשק | Web | אפליקציה | web |
| זיהוי צמחים | כן | לא | לא |
| פונקציונליות | השקיה, תאורה וצילום הצמח בעזרת מצלמת הרשת | השקיה בלבד | השקיה, זריעה וניכוש עשבים |
| אילוצי גודל | כל בקר פיזי תומך בטיפול של עד 2 עציצי בית. המערכת תומכת במספר בלתי מוגבל של בקרים. | יכולת שליטה של עציץ בית אחד בלבד. | גינה ביתית בגודל של 1mx1m. |

1. תוצר סופי
   1. תיאור הפרויקט

נושא הפרויקט הוא מערכת גידול צמח ביתי אוטונומית המאפשרת לבעלי הצמחים לבקר על גדילת הצמח מרחוק. המערכת הינה מערכת מסוג client-server, הכוללת שרת ((Server ושני סוגי clients: Plant Client ו User Client.

* Plant Client – מורכב מבקר הצמח (בקר Arduino ורכיבי I/O) ומחשב, המנהל תקשורת מול הבקר, מול השרת ומול המשתמש. למחשב מצלמה הנועדה לשידור תמונות למשתמש ולטובת איסוף נתונים על התפתחות הצמח.
* User Client – ממשק web המאפשר למשתמש לגשת לצמח מרחוק ולדוחותיו.

תפקיד ה **Plant Client** הוא זיהוי ובקרת הצמח. ה Plant Client מורכב ממערכת בקרה האוספת נתונים מחיישני לחות וחיישני תאורה ומבצעת פעולות של השקיה ותאורה. מערכת בקרה זו מחוברת למחשב רשת הכוללת מצלמת רשת. במחשב קיימת תכנית אשר תפקידה היא להחליט את אופן טיפול הצמח, כלומר כמה תאורה לספק לצמח וכמות המים הנדרשת לצמח. ההחלטות מתקבלות על בסיס מידע שמתקבל מהבקר (כמפורט לעיל) ובנוסף על פי אלגוריתם ייחודי המותאם לסוג הצמח. הPlant Client יזהה את מספר הצמחים הקיימים בעזרת מצלמת הרשת המחוברת למחשב. הזיהוי יתבצע על פי מודל בינה מלאכותית קיים (open source). זיהוי סוג הצמח יתקיים בשרת (כמפורט בהמשך) ויאפשר למערכת להתאים את אופן הטיפול בצמח למתן צמיחה אופטימלית. בנוסף על כך, ה Plant Client אחראי על איסוף מידע לצורך הפקת דוחות כגון רמת תאורה, רמת לחות, גדילת הצמח ופעולות המערכת. דוחות אלו ישלחו אל השרת ויאוחסנו שם.

התווספות צמח חדש ושיוכו לחשבון המשתמש מתבצעת על פי דרישת המשתמש, דרך ממשק הPlant Client. בתהליך זה, הצמחים נרשמים בשרת על פי סוגם ושם שניתן להם על ידי המשתמש.

במערכת הבקרה, קיימים מספר רכיבים המאפשרים טיפול בצמח:

* בקר Arduino – מנהל ומרכז את כל החיישנים ורכיבי הפלט.
* תאורה – רכיב תאורת LED, הרכיב מאיר באור צבע סגול מאחר וצבע אור זה מאפשר גדילה אופטימלית של צמחים.
* השקיה – משאבת מים המחוברת לצמח בעזרת צינור.
* מד לחות – חיישן המחובר לאדמת הצמח ומודד את לחות האדמה.
* מד אור - חיישן יחיד ומודד את עוצמת האור.
* רכיבים נוספים – משמשים לטובת מימושו של מוצר זה (מתגים, נגדים, חיווטים ולוחות).

הלוח בו אשתמש בפרויקט הינו לוח **Arduino**. הלוח מוכר בעולם בזכות סביבת פיתוח קלה ונוחה לשימוש. הבקר מקבל קלט מרכיבי חומרה וחיישנים שונים. התוכנה אשר נכתבת על ידי המשתמש וצרובה על הבקר בשפת C++ מחליטה האם ואיך להפעיל פלט מסוים (כגון נורה, מנוע, מסך).

ה **User Client** הינו ממשק המשתמש של בעל המערכת והצמחייה. רכיב מערכת זה מאפשר לבעל מערכת הצמח לגשת אל ה Plant Client מרחוק ולבצע פעולות טיפוליות כרצונו. הממשק יאפשר כניסה והרשמה של חשבונות בשרת וגישה אל דוחותיו הקיימים בשרת. קיימת אפשרות לצרף משתמש נוסף לאותה מערכת קיימת על ידי הרשמת חשבון נוסף על ידי הזנת קוד שמתקבל מאותו משתמש. לאחר ההרשמה, המשתמש יוכל להתחבר אל ה Plant Client מרחוק ובכך לבצע פעולות של השקיה ו/או תאורה ולקבל פידבק (שידור) חזותי של הצמחים בזמן אמת. בכוחו של המשתמש להחליט האם הטיפול בצמח יהיה אוטומטי או ידני. כמו כן, ה User Client מאפשר למשתמש לגשת אל הדוחות שה Plant Client יצר (ונשמרו בשרת) על פי תאריך וזמן. התראות טיפוליות חריגות של המערכת ירשמו בדוחות המעקב וישלחו ל User Client ברגע שיתחבר לחשבונו.

השרת ((**Server** מרכז את כל שלושת מסדי הנתונים וגישת המשתמש אליהם.

1. משתמשים – מסד זה כולל בתוכו את כל נתוני המשתמשים הרשומים – מספר id, שם משתמש, סיסמא (מוצפנת), צמחיהם.
2. מצבי צמח אופטימליים – מסד זה כולל בתוכו מידע על מצב הלחות, רמת האור ושעות האור המיטביים עבור צמחים שונים.
3. דוחות – מסד זה בנוי ב MongoDB בשונה משאר מסדי הנתונים מאחר והוא מאפשר עדכון והוספה של נתונים באופן מהיר יותר מאשר SQL.

בנוסף על כך, השרת מטפל בבקשות ה web מהמשתמשים בהתבסס על ספריית Flask.

* 1. אלגוריתמים עיקריים

להלן האלגוריתמים המרכזיים בפרויקט שלי:

1. חיבור וניתוק משתמשים מהמערכת
   * ה User Client שולח בקשת התחברות לחשבונו לשרת עם שם המשתמש וסיסמא המוצפנים בהצפנה חד כיוונית (hash).
   * לאחר אימות הנתונים בשרת מול מסד הנתונים, השרת שולח חזרה את המידע של חשבון המשתמש. והוא רשאי לבצע התחברות למערכת הצמח מרחוק (Plant Client). מקובל להוסיף TOKEN משתנה.
   * ה User Client שולח בקשת התחברות למערכת לשרת, השרת מאשר ומעביר ל Plant Client את הבקשה.
   * כל פקודה של ה User Client עוברת דרך השרת מלבד בקשת תמונות, העוברת ישירות ל User Client.
   * בבקשת התנתקות של ה User Client, נשלחת בקשה אל השרת ולאחר אישורו, הבקשה נשלחת מהשרת אל ה Plant Client. לאחר קבלת הבקשה ב Plant Client, הוא מפסיק להאזין לבקשות המשתמש ומנתק את העברת תמונות המצלמה.
2. שליטה בבקר

ל User Client יכולת לשלוט באופן ידני על בקר הצמח. ניתן להפסיק את הבקרה האוטומטית על הצמחים ופעולות טיפוליות מרחוק (השקיה ותאורה). כל הפקודות העוברות דרך השרת ול Plant Client מועברות בפרוטוקול TCP מאחר ופרוטוקול זה אמין ומאפשר ביטחון שהפקודה תתבצע כראוי.

* + שליחת הפקודה מה User Client אל השרת.
  + לאחר אישור השרת, הפקודה תישלח אל ה Plant Client.
  + ב Plant Client, הפקודה מומרת ומועברת דרך חיבור USB אל הבקר.

1. אלגוריתם זיהוי הצמח
   * ה Plant Client מזהה את מספר הצמחים ומצלם כל צמח בנפרד בעזרת בינה מלאכותית.
   * כל אחת מהתמונות נשלחות בנפרד לצורך זיהוי סוג הצמח.
     + הזיהוי מתבצע בעזרת ה API של נותן השירות Plant.id – בעזרת API זה, המערכת מסוגלת לזהות את סוג הצמח בעזרת תמונה יחידה.
   * לאחר ניתוח התמונה, השרת מחזיר תשובה מפוענחת המפרטת את מספר הצמחים, סוגי הצמחים ואת התנאים לטיפול האופטימליים של הצמחים הקיימים במסד הנתונים.
   * אילו הזיהוי אינו מספק תשובה חד משמעית, לא יתקיים טיפול אוטומטי בצמח והשרת יבקש מהמשתמש להזין את סוג הצמח באופן ידני. מנהלי המערכת יקבלו התראה על מנת שיוסיפו את המידע באופן ידני.
   * אילו לא קיים מידע על הצמח המפוענח במסד הנתונים, לא יתקיים טיפול באותו צמח. מנהלי המערכת יקבלו התראה על מנת שיוסיפו את המידע באופן ידני.
   * ה Plant Client מצלם תמונה של המערכת בעזרת המצלמה המחוברת אליו.
   * התמונה נשלחת אל השרת לניתוח. אחת מספיק? ניתוח תמונות הצמח מתבצע בעזרת ספריית PlantCV. PlantCVהיא ספרייה לניתוח תמונות צמחים הכוללת מספר מודלים של למידת מכונה מאומנים מראש שניתן להשתמש בהם כדי לסווג צמחים על פי תמונות הצמח וגודלם.
   * לאחר ניתוח התמונה, השרת מחזיר תשובה מפוענחת המפרטת את מספר הצמחים, סוגי הצמחים ואת התנאים לטיפול האופטימליים של הצמחים הקיימים במסד הנתונים. מה קןרה אם לא מתקבלת תשובה? האם התשובה היא סטטיסטית? כלומר בסיכוי X זה Y וכו'? מה קורה אם יש נתונים גבוליים?
   * במידה ולא קיים מידע על הצמח המפוענח במסד הנתונים, לא יתקיים טיפול באותו צמח. מנהלי המערכת יקבלו התראה על מנת שיוסיפו את המידע באופן ידני. איך מוסיפים ידני? כמו כן, המשתמש יקבל התראה על כך שהצמח אינו מטופל באופן אוטומטי. מסכים שלא אוטומטי, אבל האמירה לא יתקיים טיפול?
2. אלגוריתם טיפול הצמח

בהתאם לנתוני הצמח האופטימליים המוגדרים במסד הנתונים, ולפי המידע המגיע מחיישני התאורה והלחות, תתבצע בקרת השקיה והדלקת תאורה.

* + שגרת איסוף נתונים - הטיפול בצמח מתבצע בצורה מחזורית. בכל פרק זמן קבוע (לדוגמא 5 דקות משעות הבוקר המוקדמות עד שעות הערב המאוחרות), מתבצע איסוף נתוני האור והלחות בעזרת החיישנים.
  + השקיה (עקרון Hysteresis) -
    - ההשקיה תתבצע כאשר מד הלחות ירד מסף המינימום בפולסים קטנים על מנת להימנע ממצב של השקית יתר.
    - מדידת הלחות תתבצע לאחר כל פולס עד אשר מד הלחות יעלה על סף המקסימום.
    - חזרה לשגרה.
  + תאורה –
    - הדלקת התאורה תתבצע בשעות השקיעה ותתחשב בכמות זמן האור שנצבר במהלך היום עד להגעת כמות שעות האור האופטימליות.

1. אלגוריתם שיפור המידע בנוגע למצב המיטבי של זן צמח (למידה חישובית)
   * לקוחות וותיקים יוכרזו כלקוחות נסיינים על ידי השרת. לקוחות אלו יקבלו את נתוני טיפול הצמח (תאורה והשקיה) השונים במעט מהנתונים האופטימליים הקיימים במסד הנתונים. לדוגמא, צמח הצורך 5 שעות תאורה ביום, יטופל בתוספת של חצי שעת תאורה. או פחות? האם יוגדרו מספר קבוצות ניסוי?
   * מעקב הגדילה יתבצע על ידי תמונות הצמח לאורך תקופת מוגדרת (לדוגמא שבועיים). חישוב אחוז גדילת הצמח מתבצע בעזרת אלגוריתם זיהוי הצמח מיום תחילת הניסוי.
   * לאחר קיבוץ המידע בשרת, מתבצע אלגוריתם מבוסס bayes על מנת להחליט האם לשנות את תנאי הגידול של הצמח לטובת שיפור קצב גדילתו.
   * במידה והתרחש שינוי מידע, השרת ישלח לכל המשתמשים בעלי אותו סוג צמח את התנאים החדשים והמשופרים.
2. יצירת דוחות

ה Plant Client מבצע איסוף מידע רב של הצמח וסביבת גדילתו.

* + שם הצמח.
  + זמן איסוף המידע.
  + המידע מהחיישנים (כפי שתואר בסעיף טיפול הצמח).
  + זמן ביצוע הפעולה ועל ידי מי (אוטומטי / משתמש).
  + זמני התחברות ה User Client אל ה Plant Client.
  + חישוב אחוז הגדילה של הצמח בעזרת המצלמה (פעם ביום).

הדוחות יישלחו לשרת בסוף כל יום ויישמרו במסד הנתונים. המידע יוצג אצל המשתמש באופן גרפי וידידותי.

* 1. דרישות ואילוצי פתרון

לפרויקט מספר דרישות:

* נדרשת תמיכה באופן גמיש במספר הלקוחות אשר מתחברים אל השרת בו זמנית.
* נדרשת תמיכה בשיפור ושינוי המידע הטיפולי על פי נתוני השטח מלקוחותיו השונים. השינויים מתבצעים בעזרת למידה חישובית.
* נדרשת תמיכה בהוספת/הסרת צמחים למערכת ולחשבון המשתמש.
* נדרשת תמיכה של 2 חשבונות המקושרים לאותו Plant Client.

לפרויקט מספר אילוצים:

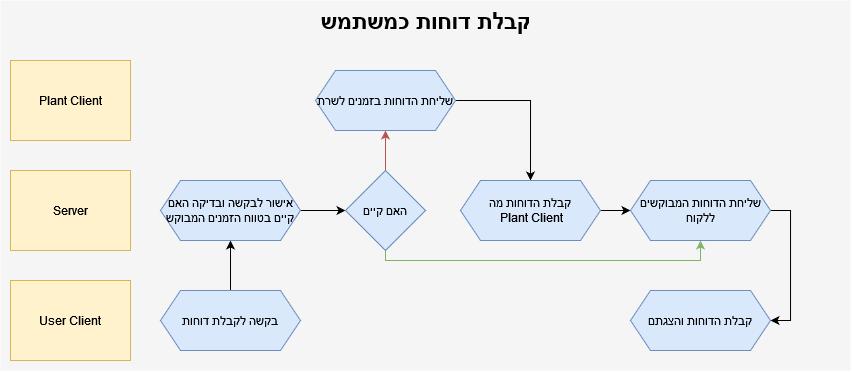
* בקר יחיד תומך בניהול של עד 2 צמחים לאור אילוציו הטכניים.
* למערכת אין אפשרות להתריע למשתמש כאשר מי ההשקיה וסוללות המשאבה נגמרו ואחד הרכיבים התקלקל עקב אילוצי חומרה טכניים.
  1. ממשקים למערכות חיצוניות

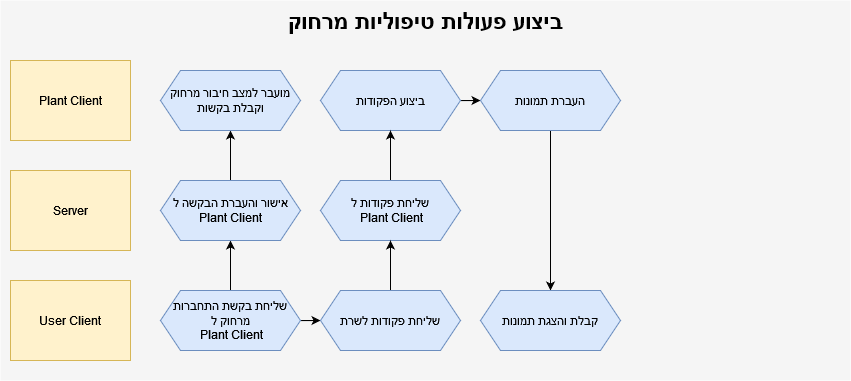
בהמשך

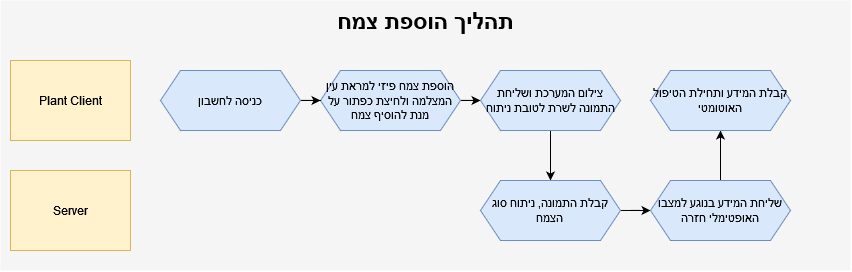
* 1. התייחסות לנושאי אבטחה

בהמשך

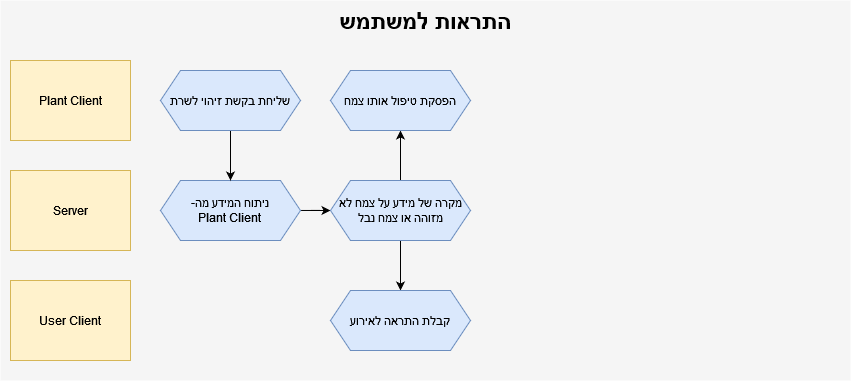
* 1. תרחישים עיקריים







יפה וגם הוספת משתמש!



* 1. ממשק משתמש

להשלים

1. תהליך כתיבת הפרויקט
   1. תהליך הפרויקט

כאשר הוטל עלינו לבחור את הנושא לפרויקט הגמר של כיתה יב', ידעתי שאני רוצה ליצור משהו חדשני ושונה שיחולל מהפכה באופן שבו אנשים מטפלים בצמחים שלהם. באופן אישי, אני מגדל מספר צמחים בבית ונהנה להשקות אותם ולטפל בהם, אך הדבר מידי פעם מוביל לקשיים כאשר אני נאלץ להיות מחוץ לבית תקופה ממושכת. לכן, מתוך צורך אישי, בחרתי לבנות מערכת לבקרת צמח (IoT), המסוגל לנטר את בריאות הצמח ולהתאים את סביבתו כדי לספק תנאים אופטימליים לצמיחה.

התחלתי בחקר חיישנים שונים שניתן להשתמש בהם כדי למדוד את לחות הקרקע, ורמות האור. לאחר קביעת החיישנים המתאימים לשימוש, בחרתי את האלמנטים דרכם אבצע את הטיפול הפיזי בצמח – משאבת המים ותאורת ה-LED. לבסוף, תכננתי את ארכיטקטורת המערכת שתאפשר ניטור בזמן אמת של מצב הצמח וטיפול מידי.

השלב הבא היה תכנון מערכת הבקרה שתנהל את סביבת המערכת. החלטתי להשתמש ב-Arduino בתור הבקר של המערכת, דבר היאפשר אינטגרציה קלה עם החיישנים, האלמנטים והתקשורת מול מחשב. כחלק מה-POC (Proof Of Concept), לאחר מכן תכנתתי את הבקר לאסוף נתונים מהחיישנים, להפעיל את הרכיבים המתאימים, וביצוע שליטה מרחוק של הבקר בזמן אמת.

אחד האתגרים הגדולים ביותר שעמדתי בפני היה עיצוב ממשק משתמש שיהיה גם אינטואיטיבי וגם מושך מבחינה ויזואלית. רציתי שממשק המשתמש יהיה קל לשימוש, אך עדיין יספק את כל המידע הדרוש על בריאות הצמח ומצבו. לאחר התנסות בכמה רעיונות עיצוב שונים, וקבלת חוות דעה מהמורה ואנשים שונים, התקבעתי על ממשק אלגנטי ומודרני.

כשעבדתי על הפרויקט, הבנתי שכתיבת פרויקט בהיקף ובמורכבות כאלה דורשת מערך מיומנויות שונה מאשר כתיבת תוכנית קטנה ופשוטה. הייתי צריך לחשוב על כל ארכיטקטורת המערכת וכיצד כל הרכיבים יתקשרו זה עם זה עוד לפני תחילת כתיבת הפרויקט. בנוסף על כך, הייתי צריך גם לקחת בחשבון נושאים כמו אבטחה ופרטיות נתונים, שהם קריטיים עבור כל מערכת IoT.

למרות האתגרים, מצאתי שהפרויקט מתגמל להפליא. לא רק שרכשתי מיומנויות טכניות חשובות ב-IoT ועיצוב מערכות, אלא גם למדתי הרבה על ניהול וכתיבה של פרויקטים. כעת אני מרגיש בטוח יותר ביכולת שלי לקחת על עצמי פרויקטים בקנה מידה גדול ולספק תוצאות באיכות גבוהה.

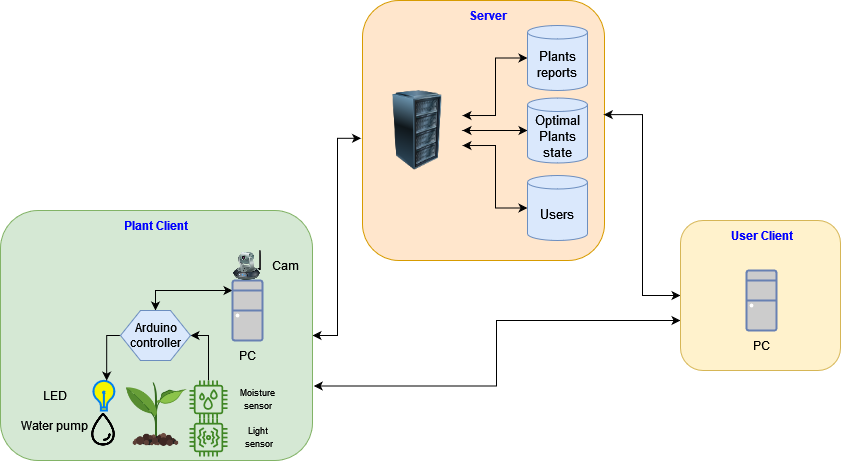
* 1. אתגרים ואופציות שונות למימוש

חלק מהאתגרים שעמדתי בפני בפרויקט זה היו:

* שילוב חומרה ותוכנה: לא היה קל לשלב רכיבי חומרה שונים ולגרום להם לעבוד בצורה חלקה עם התוכנה של המחשב. נאלצתי להתמודד עם בעיות כמו תאימות, תקשורת ואספקת חשמל. הייתי צריך לבדוק ולנפות באגים ברכיבי החומרה והתוכנה באופן נרחב כדי להבטיח את הפונקציונליות והאמינות שלהם. פתרתי את האתגר בעזרת למידה והבנה עמוקה של דרכי התקשורת האפשריים בין הבקר לבין המחשב, החלטתי לתקשר בינהם בעזרת Serial port – החיבור מאפשרת תקשורת בין קוד Python הפועל על המחשב ללוח Arduino באמצעות חיבור USB. דבר זה מאפשר לתוכנית ה Python לשלוח פקודות לבקר ולקבל ממנו נתונים.
* תכנון ויישום: התכנון והיישום של המערכת שיכולה לשלוט בצמיחת הצמחים מרחוק היה תהליך מורכב שדרש תכנון, קידוד ובדיקה קפדנית. הייתי צריך להבטיח שהמערכת תהיה ידידותית למשתמש, יעילה ואמינה. הייתי צריך לעקוב אחר מחזור החיים של פיתוח התוכנה ולהשתמש בכלים ומתודולוגיות מתאימות.
* פיתוח אלגוריתם: אחד האתגרים המרכזיים היה לפתח אלגוריתם שיוכל לזהות במדויק את סוג הצמח ולהתאים את התנהגותו בהתאם. זה דרש ידע מעמיק בלמידת מכונה, בינה מלאכותית וראייה ממוחשבת. הייתי צריך לאמן ולבדוק את האלגוריתם באמצעות מערכי נתונים שונים ולהעריך את הביצועים והדיוק שלו.
* אבטחה: מכיוון שהמערכת כללה גישה ושליטה מרחוק, הבטחת אבטחת המערכת והנתונים הייתה קריטית. הייתי צריך לאבטח את התקשורת בין השרת ללקוחות, להגן על נתוני משתמשים ולמנוע גישה לא מורשית. הייתי צריך להשתמש בטכניקות הצפנה, אימות והרשאה כדי להגן על המערכת.
* ניהול נתונים: אתגר נוסף היה לאסוף ולנהל נתונים מחיישנים ורכיבים שונים. זה היה מאתגר, במיוחד כאשר התמודדו עם מספר רב של צמחים. הייתי צריך לוודא שהנתונים מדויקים, עקביים ומאוחסנים בצורה מאובטחת. הייתי צריך גם לנתח ולפרש את הנתונים כדי ליצור תובנות ודוחות שימושיים.
* מדרגיות: ככל שהמערכת גדלה, התמודדתי עם אתגרים בהגדלת המערכת, הוספת מפעלים חדשים והכלת משתמשים נוספים. הייתי צריך לעצב את המערכת מתוך מחשבה על מדרגיות ולהבטיח שהיא תוכל להתמודד עם הגדלת נפחי התעבורה והנתונים. הייתי צריך להשתמש במחשוב ענן, איזון עומסים וטכניקות שמירה במטמון כדי לשפר את המדרגיות של המערכת.

1. מרכיבי פתרון
   1. תיחום הפרויקט

* תקשורת – תקשורת בפרוטוקול TCP של הלקוח מול השרת. כמו כן, קיימת ביקורת על הפעולות המבוצעות מרחוק.
* אבטחת מידע – רישום וכניסת משתמשים בצורה מאובטחת אל המערכת על פי שם משתמש וסיסמא.
* מערכת הפעלה – שימוש נרחב ב-Threads.
* תצוגה – גרפיקת web מרשימה וידידותית לשני סוגי הלקוחות בעזרת ספריית Flask.
* מבנה נתונים – שימוש רב במבני נתונים רבים כגון רשימות, מילונים ומשתנים וקבועים על מנת להקל על כתיבת הקוד בצורה היעילה ביותר.
* אלקטרוניקה - בקר Arduino ורכיבים אלקטרונים נוספים לטובת בקרת טיפול הצמח.
* בסיסי נתונים - שרתי SQL של המשתמשים הרשומים ונתוני הצמח האופטימליים של זני צמחים מסוימים היות ולשפת תכנות זו יכולת ביצוע שאילתות מתקדמות. דוחות הבקרה של הצמחים הפעילים יאוכסנו מסד נתונים מסוג NOSQL מכיוון שמסד נתונים זה בעל יכולות עדכון ויצירת מידע מהירים לעומת SQL.
* שימוש במחלקות – שימוש מתקדם במחלקות לטובת הקלת כתיבת הקוד.
* תיעוד – תיעוד לכל בלוק של קוד הדורש זאת, כתיבת ספר פרויקט תוך הליך העבודה ופתיחת GIT לטובת מעקב אחר גרסאות.
  1. סביבת העבודה (טכנולוגיה)
* שפות התכנות:
  + Python – לקוחות ושרת
  + C++ - שפת בקר ה Arduino
* סביבות פיתוח:
  + Pycharm – לשפת Python
  + Visual Studio Code – לכתיבת HTML ו JavaScript
  + Atom – לשפת c++ והעלאת הקוד לבקר
  1. מבט טופולוגי



יפה, ויש יותר מ USER אחד

ולהסביר, בהמשך

* 1. מבנה נתונים

בהמשך

* 1. מסד נתונים

בהמשך

* 1. מבט מודולרי

בהמשך

* 1. פירוט מודלים עיקריים

להלן הפעולות המרכזיות והחשובות ביישום הפרויקט:

|  |  |  |  |
| --- | --- | --- | --- |
| **מחלקה** | **פעולה** | **פלט וקלט** | **תיאור** |
| ArduinoRobot | reconnect\_board | Input: baud(int), timeout(int) Output: bool | ניסיון חיבור מחדש ללוח Arduino, תוך שימוש באותם או ערכי באוד ופסק זמן חדשים. |
| send\_and\_receive | Input: msg(str), rec(bool) Output: str/None | שולח הודעה ללוח Arduino המחובר, ממתין לתגובה אם rec=True, ומחזיר את התגובה אם לא מזוהה "ERROR" בהודעה. |
| set\_text\_display | Input: msg(str), rec(bool) Output: str | מגדיר את הטקסט שיוצג על מסך ה-LCD המחובר ל-40 התווים הראשונים של ההודעה הנתונה. |
| get\_moisture\_level | Input: plant(str), rec(bool) Output: str | מאחזר ומחזיר את רמת הלחות של הצמח שצוין. |
| get\_light\_level | Input: plant(str), rec(bool) Output: str | מאחזר ומחזיר את רמת האור של הצמח שצוין. |
| add\_water | Input: plant(str), duration(str), rec(bool) Output: None | מפעיל את משאבת המים למשך הזמן שצוין (בשניות) עבור המפעל שצוין. |
| set\_light | Input: plant(str), mode(bool), rec(bool) Output: None | מגדיר את מצב טבעת ה-LED עבור המפעל שצוין, הפעלה או כיבוי שלה בהתאם לערך המצב. |
| PlantRecognitionManager | take\_picture | Input: purpose(str) | מצלם תמונה עם האובייקט PictureGrabber, עם ארגומנט מטרה אופציונלי. |
| run | Input: input\_image\_path(str), output\_image\_path(str), num\_plants(int), current\_plants(int) | מפעיל את תהליך זיהוי הצמח על ידי צילום תמונה, זיהוי צמחים עם אובייקט PlantDetector, עיבוד הצמחים שזוהו ורישום נתוני הצמח בשרת. |
| process\_detected\_plants | Input: detect(bool) | מעבד את הצמחים שזוהו על ידי שליחת התמונה למטפל השרת, חילוץ נתוני הצמח ורישום הצמח אצל השרת. |
| EventLogger | send\_event | Input: cevent(tuple) Output: bool | Sends an event to the server using the server handler object and returns a boolean indicating success. |
|  | send\_growth\_event | Input: cevent(tuple) Output: bool | Sends a growth event to the server using the server handler object and returns a boolean indicating success. |
|  | add\_auto\_action\_event | Input: user\_id(str), level(str), action(str), send\_now(bool) Output: None | Creates an action event with the given user ID, level, and action, and writes it to a file using the write\_event() method. If send\_now is True, also sends the event to the server. |
|  | add\_data\_event | Input: user\_id(str), moisture(float), light\_lvl(float), light\_hours(float) Output: None | Creates a data event with the given user ID, moisture level, light level, and light hours, and writes it to a file using the write\_event() method. |
|  | add\_growth\_event | Input: user\_id(str), plant\_name(str), light\_level(float), moisture\_level(float), height\_px(int) Output: None | Creates a growth event with the given user ID, plant name, light level, moisture level, and height in pixels, and writes it to a file using the write\_event() method. If the event is a growth event, sets event\_type to "growth". |
|  | write\_event | Input: event(obj), send\_now(bool), event\_type(str) Output: None | Writes the given event object to a file in the designated directory using the pickle module. If send\_now is True, also sends the event to the server using the appropriate method depending on the event type (either send\_event() or send\_growth\_event()). |
|  | remote\_event\_logger | Input: user\_id(str), action\_data(str), send\_now(bool), threaded(bool) Output: None | Creates an action event with the given user ID, "Manual" level, and action data, and either writes it to a file using the write\_event() method and sends it to the server (if send\_now is True and threaded is False) or starts a new thread to perform these actions (if threaded is True). |
|  | automatic\_event\_logger | Input: user\_id(str), action\_data(str), send\_now(bool), threaded(bool) Output: None | Creates an action event with the given user ID, "Automatic" level, and action data, and either writes it to a file using the write\_event() method and sends it to the server (if send\_now is True and threaded is False) or starts a new thread to perform these actions (if threaded is True). |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. תסריטי בדיקה
   1. דגשים בבדיקה

הדגשים שלי בבדיקה היו:

בהמשך

* 1. תסריטי בדיקה עיקריים

בהמשך

1. רפלקציה
   1. לוח זמנים מוערך לניהול הפרויקט:

|  |  |
| --- | --- |
| נובמבר | POC הכולל remote control על בקר ה-Arduino מרחוק דרך השרת. הפעולות אפשרו למשתמש לבצע פעולות ולקרוא מידע מהחיישנים המחוברים לבקר בזמן אמת. |
| דצמבר | כניסת והרשמת משתמשים, שליטה על בקר ה-Arduino והפונקציות השונות שלו מרחוק, הגדרת אופן טיפול (אוטומטי/ידני), הפקת דוחות הקשורים לפעולות השונות אשר בוצעו באופן אוטומטי או ידני וניהול שלושת מסדי נתונים למשתמשים, למצב הצמחים האופטימלי והדוחות. |
| ינואר | מערכת Arduino סופית הכוללת את כל הרכיבים והחיישנים, זיהוי הצמח בעזרת בינה מלאכותית, שידור תמונה למשתמש מרחוק (E2E) והפקת דוחות מידע על גדילתו של הצמח. |
| פברואר | אלגוריתם הטיפול בצמח הנשען על סוג הצמח והמידע מהמשתמשים. |
| מרץ | גרפיקה מרשימה וידידותית למשתמש, אימות של מבצע הפעולות מרחוק על מנת למנוע חבלה, בקרת שגיאות, מוצר וספר סופיים. |

* 1. אתגרים ותרומה אישית

אתגרים:

* רכישת סט מיומנויות חדש: כשהתחלתי לעבוד על הפרויקט, אחד האתגרים הגדולים ביותר שעמדתי בפני היה ללמוד מערך מיומנויות חדש שלא היה לי ניסיון קודם בה. זה היה מרתיע לחשוב כמה אני צריך ללמוד בכמות קצרה של זמן להיות מסוגל לספק מוצר איכותי.
* ניהול זמן: ניהול הזמן שלי בצורה יעילה היה אתגר נוסף שנתקלתי בו במהלך הפרויקט. היו לי התחייבויות רבות, והאיזון ביניהן עם הפרויקט היה משימה קשה. הייתי צריך לתעדף את העבודה שלי ולקבוע לוח זמנים ריאלי כדי להבטיח שאוכל להשלים את הפרויקט בזמן.
* התמודדות עם כישלונות: במהלך הפרויקט התמודדתי עם כמה כישלונות שאיימו לדרדר את ההתקדמות שלי. זה היה מאתגר לשמור על מוטיבציה ולדחוף את הכישלונות האלה, אבל למדתי להיות גמיש ולהסתגל לנסיבות משתנות.

תרומה אישית:

* מחויבות: הייתי מחויב לספק מוצר איכותי, ודאגתי להשקיע את המאמץ והזמן הנדרשים כדי להשיג זאת. עבדתי בעקביות ובשקידה, גם כשזה היה אומר להקריב חלק מזמני הפנאי שלי.
* יצירתיות: הבאתי גישה יצירתית לפרויקט, שעזרה לי להעלות רעיונות ופתרונות חדשניים. תמיד חשבתי מחוץ לקופסה וחיפשתי דרכים להפוך את הפרויקט ליותר מרתק ומשפיע
* גישה שיתופית: בעוד שהפרויקט היה בעיקרו מאמץ אינדיבידואלי, הקפדתי לחפש משוב ותשומות מאחרים בכל פעם שיכולתי. שיתפתי פעולה עם עמיתים ומנטורים, והתובנות שלהם עזרו לי לשפר את הפרויקט בדרכים משמעותיות
* התמדה: למרות האתגרים שעמדתי בפני, התמדתי ונשארתי ממוקד במטרות שלי. סירבתי לוותר, גם כשנדמה היה שהפרויקט מכריע או בלתי אפשרי להשלים.
  1. תובנות

תכנון, כתיבת וביצוע הפרויקט היה חוויה מאתגרת אך מתגמלת שעזרה לי להרחיב את ההבנה שלי במושגי מחשב שונים, חדשים וישנים כאחד. ככל שהתקדמתי ופיתחתי את הפרויקט, לא רק למדתי על ההיבטים הטכניים של כתיבת פרויקט שכזה אלא גם קיבלתי תובנה לגבי החשיבות של שמירה על יעדי הפרויקט ארוכי טווח והיכולת לבצע שינויים נחוצים בשלבי התכנון והבנייה.

הרעיון לפרויקט הזה עלה בי כי רציתי ליצור משהו שיועיל לאנשים כמוני – אנשים בעלי צמחיה ביתית. עם זאת, ככל שהפרויקט התפתח, התחלתי לדאוג שהקונספט העיקרי שתכננתי בהתחלה לא יעמוד ביעדי הפרויקט המקוריים שלי. למרות החששות שלי, התמדתי, ובסופו של דבר, הפרויקט התברר כייצוג אמיתי של החזון שלי, אם כי עם כמה שינויים בדרך.

בתחילת הפרויקט התמלאתי במוטיבציה, תקווה ואמונה שאצליח להגשים את מה שהתכוונתי לעשות. עם זאת, ככל שהתקדמתי, הבנתי כמה עבודה והשקעה נדרשת כדי להביא את הפרויקט לפועל. היו רגעים של תסכול וייאוש, אבל המוטיבציה שקבעתי לעצמי החזיקה אותי. במבט לאחור, אני שמח שלא ויתרתי על הפרויקט. אני גאה במה שהשגתי, ואני יכול לראות כמה גדלתי והתפתחתי כאדם ומתכנת לאורך כל התהליך.

אחד הלקחים המשמעותיים ביותר שלמדתי מניסיון זה הוא חשיבות התכנון והארגון בעת התמודדות עם בעיות מורכבות. על מנת להבטיח שהמוצר הסופי שלי עונה על הציפיות שלי, הייתי צריך לקחת את הזמן כדי למפות ביסודיות כל שלב בתהליך ולהתאים את התוכניות שלי בהתאם. רמה זו של תשומת לב לפרטים הוכחה כחיונית בכל תחומי חיי, הן בתוך ומחוץ לחיי הבית ספר.

לסיכום, ביצוע הפרויקט הזה הראה לי שעם מספיק מוטיבציה ונחישות, אין אתגר שאי אפשר להתגבר עליו. צברתי ביטחון ביכולות שלי, ואני מצפה לקחת על עצמי פרויקטים חדשים בעתיד.

1. הוראות התקנה ותפעול
   1. תצורה ודרישות קדם

למחשב ה Server:

* + חיבור לאינטרנט
  + Python 3.8+
  + ספריות – cv2, flask, pickle, threading, hashlib, json, sqlite3, os, base64, zlib, pymongo, matplotlib
  + מסד נתונים SQL במבנה המתאים
  + מסד נתונים MongoDB הרץ בכתובת mongodb://localhost:27017

למחשב ה Plant Client:

* + חיבור לרשת
  + Python 3.8+
  + בקר Arduino ורכיביו
  + חיבור USB פנוי
  + מצלמת רשת מובנת/חיצונית
  + ספריות – serial, pickle, os, threading, PIL, cv2, re, sched, tkinter, imageai, zlib, flask, base64

למחשב ה User Client:

* + חיבור לרשת
  1. התקנה

לאחר התקנת הדרישות, הספריות והתקנת המערכת של הצמח, יש להפעיל את השרת הראשי. לאחר מכן כל משתמש יכול להתחבר לשרת וללקוח שלו מתי שירצה ללא תלות אחד בשני.

* 1. הפעלה

בהמשך

1. ביבליוגרפיה

במהלך כתיבת הפרויקט הסתמכתי על מספר מקורות מידע:

1. נספחים

בהמשך