A logo of a book with a atom and blue text

Description automatically generated

פלטפורמת למידה וניהול סטודנטים מתקדמת

**מחזור נ"ט – מסלול הנדסאי תכנה**

**המכללה הטכנולוגית רופין**

**מגישי הפרויקט:**

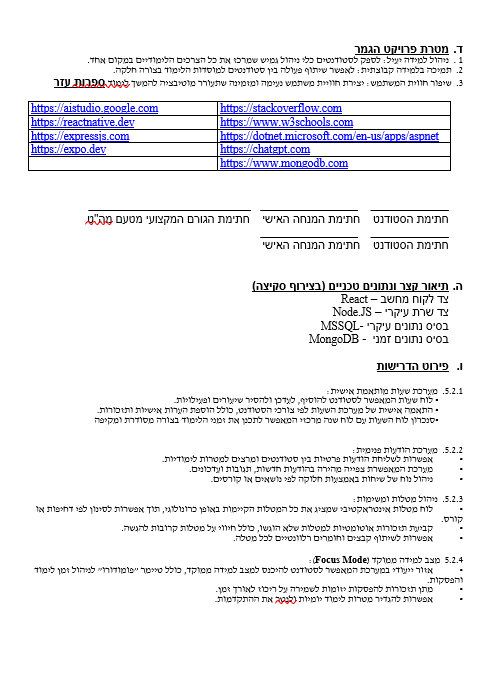
o סרגיי ביררוב

o מיכאל קולומיזה

2024

**A document with text and numbers

Description automatically generated**

****

****

****

**A document with a signature

Description automatically generated**

A black background with blue and green text

Description automatically generated

**(2.1) אקדמיק – המלווה האולטימטיבי לסטודנטים ולמוסדות לימוד אר מכיל מגוון כלים שיכוונו את הסטודנט ואת מוסדות הלימוד להישגים אקדמאיים ולניהול זמנים נכון ואפקטיבי.**

**המערכת מקדמת ומרכזת את מטרות הסטודנט לעמוד בדרישות התואר או המסלול בו לומדים תוך הפחתה בלחצים ממבחנים קרבים, מטלות ופרויקטים להגשה על ידי קיזוז המסע האקדמי לחוויה מאורגנת ויעילה שתעזור לנהל את החומרים הנדרשים, קורסים, כלים לשיתוף פעולה וייעול זמנים שתעזור לסטודנט להגשים את הפוטנציאל החבוי בו.**

## (2.2) ממטרות המערכת

**1. ניהול למידה יעיל: לספק לסטודנטים כלי ניהול גמיש שמרכז את כל הצרכים הלימודיים במקום אחד.**

**2. תמיכה בלמידה קבוצתית: לאפשר שיתוף פעולה בין סטודנטים למוסדות הלימוד בצורה חלקה.**

**3. שיפור חווית המשתמש: יצירת חוויית משתמש נעימה ומזמינה שתעורר מוטיבציה להמשך לימוד.**

## **(3.1)** המצב בשוק אותו אנו שואפים לתקן

**כיום קיימות בשוק מערכות לימוד רבות כגון "ידיעון". מערכות אלה מספקות שירותי בסיסיים לניהול הלמידה כמו ניהול מטלות, קורסים, וציון הישגים.**

## (3.2) בעיות במצב הקיים

**3.2.1 ממשק משתמש מסורבל: רוב המערכות הקיימות לא מספקות חווית משתמש נעימה ויעילה. הן לוקות בעיצוב שאינו מזמין ושאינו מסייע לסטודנטים לניהול הלמידה.**

**3.2.2 חוסר סדר וארגון: אין פתרון מובנה לסידור המטלות והמשימות באופן כרונולוגי או על פי דחיפות.**

**3.2.3 חוסר סנכרון בין מערכות: אין סנכרון בין לוח הזמנים של הקורסים לבין לוח השנה, דבר המוביל לפספוס משימות**

## (4) מה באנו לחדש או לשפר

**4.1. שיפור ויזואלי ושימושיות: יצירת ממשק משתמש אינטואיטיבי ומושך לשיפור חווית השימוש של הסטודנט.**

**4.2. כלים ייחודיים לניהול זמן ולמידה: פיתוח כלים לניהול זמן כמו טיימר "פומודרו" ושילוב תזכורות מותאמות אישית.**

**4.3. שיפור אינטגרציה: יצירת סנכרון מלא בין כל חלקי המערכת, כגון לוח שנה, מטלות, ומסלול הלימודים.**

## (5) מדרישות המערכת

**5.1. ביצועים גבוהים: על המערכת לספק חווית שימוש רציפה ללא תקלות או בעיות מהירות.**

**5.2. תאימות בין מכשירים: התאמת המערכת למגוון המכשירים, כולל מחשבים, טאבלטים, וסמארטפונים.**

**5.3. אבטחת מידע: הצפנת כל המידע הרגיש של הסטודנטים והמוסדות**

## (5.2) מהדרישות הפונקציונליות

**5.2.1. מערכת שעות מותאמת אישית:**

* **לוח שעות המאפשר לסטודנט להוסיף, לעדכן ולהסיר שיעורים ופעילויות.**
* **התאמה אישית של מערכת השעות לפי צורכי הסטודנט, כולל הוספת הערות אישיות ותזכורות.**
* **סנכרון לוח השעות עם לוח שנה מרכזי המאפשר לתכנן את זמני הלימוד בצורה מסודרת ומקיפה.**

**5.2.2. מערכת הודעות פנימית:**

* **אפשרות לשליחת הודעות פרטיות בין סטודנטים ומרצים למטרות לימודיות.**
* **מערכת המאפשרת צפייה מהירה בהודעות חדשות, תגובות ועדכונים.**
* **ניהול נוח של שיחות באמצעות חלוקה לפי נושאים או קורסים.**

**5.2.3. ניהול מטלות ומשימות:**

* **לוח מטלות אינטראקטיבי שמציג את כל המטלות הקיימות באופן כרונולוגי, תוך אפשרות לסינון לפי דחיפות או קורס.**
* **קביעת תזכורות אוטומטיות למטלות שלא הוגשו, כולל חיווי על מטלות קרובות להגשה.**
* **אפשרות לשיתוף קבצים וחומרים רלוונטיים לכל מטלה.**

**5.2.4 מצב למידה ממוקד (Focus Mode):**

* **אזור ייעודי במערכת המאפשר לסטודנט להיכנס למצב למידה ממוקד, כולל טיימר "פומודורו" לניהול זמן לימוד והפסקות.**
* **מתן תזכורות להפסקות יזומות לשמירה על ריכוז לאורך זמן.**
* **אפשרות להגדיר מטרות לימוד יומיות ולנטר את ההתקדמות.**

**5.2.5. מערכת ניהול חומרים וקורסים:**

* **כל קורס מזוהה על ידי מזהה ייחודי (CourseId) וכולל את שם הקורס ואת הדרישה לכיתה (ClassRoomRequired).**
* **ניתן לשייך קורסים לסטודנטים דרך טבלת ActiveStudentCourses, שמבטיחה את הקשר בין כל סטודנט לקורסים הפעילים שלו**

**.5.2.6. מעקב והנפקת דוחות התקדמות:**

* **מעקב אחרי ההתקדמות של הסטודנט בכל קורס, כולל צפייה בציונים, סטטיסטיקות של מטלות שהוגשו ונקודות לשיפור.**
* **הנפקת דוחות חודשיים או סמסטריאליים עם המלצות לשיפור, המתבססות על ההתקדמות הנוכחית.**

**5.2.7. לוח שנה אינטראקטיבי:**

* **לוח שנה משולב המציג את השיעורים, המטלות והאירועים החשובים, מסונכרן עם מערכת השעות.**
* **אפשרות להוספת אירועים אישיים (כגון מפגשי לימוד או חופשות) ולהגדיר תזכורות בהתאם.**
* **שילוב של מטלות ודדליינים בצורה ויזואלית לניהול זמן אפקטיבי.**

**5.2.8. ניהול על פי קורס:**

* **לכל קורס יש חלק במערכת שבו ניתן לצפות בכל המידע והחומרים הקשורים אליו, כולל מטלות, חומרים, ודירוגים.**
* **בעת כניסה לשיעור, המערכת תדע להציג לסטודנט את כל הכלים והחומרים הנדרשים לאותו שיעור בצורה אינטואיטיבית.**
* **שילוב כלי למידה מיוחדים לכל קורס בהתאם לצרכיו, כמו משאבים, המלצות ללימוד, ותזכורות.**

**5.2.9. צ'אט חי:**

* **אפשרות לצ'אט בזמן אמת עם סטודנטים אחרים או עם המרצה לשאלות ובירורים.**
* **יכולת להצטרף לקבוצות לימוד ולנהל שיחות קבוצתיות.**
* **ממשק נוח המאפשר חיפוש בהודעות, תיוג משתתפים, ושליחת קבצים.**

**5.2.10 ניהול זמן ואבטחת מידע:**

* **ניהול ואימות סשנים מבוסס על זמן שימוש, כך שהמשתמשים מחויבים לבצע ריענון של תהליך הזדהות לאחר פרקי זמן מוגדרים.**
* **שמירת כל המידע הרגיש מוצפן במאגרי הנתונים תוך שימוש בתקני הצפנה מתקדמים, והתחייבות לשמירת פרטיות המשתמשים.**

**5.2.11. שימוש בממשקים (API) למערכות נוספות:**

* **אינטגרציה עם כלי למידה נוספים ומערכות חיצוניות כדי להעשיר את המידע הזמין לסטודנטים.**
* **אפשרות לייבא או לייצא חומרים ממערכות שונות, ולהציג תכנים מתאימים בתוך המערכת.**

## (6) בעיות צפויות ופתרונות

**6.1.**

* + 1. **אבטחת מידע: שמירה על המידע האישי והאקדמי של המשתמשים מצריכה רמת אבטחה גבוהה ביותר.**
    2. **תאימות: התאמה למגוון מכשירים ומערכות הפעלה מצריכה פיתוח רספונסיבי.**
    3. **מורכבות ממשק משתמש: יצירת ממשק ידידותי וברור עשויה להיות מאתגרת כאשר מדובר במספר רב של תכנים ותכונות.**
  1. **) פתרונות אפשריים**
     1. **אבטחת מידע: שימוש בהצפנת JWT ובתקני הצפנה מתקדמים לשמירה על פרטיות.**
     2. **תאימות: שימוש ב־React לצורך בניית ממשק רספונסיבי המתאים לכל המכשירים.**
     3. **ממשק משתמש: שימוש במערכות עיצוב מבוססות קהילה כגון Material UI, ובדיקות משתמשים כדי לשפר את הממשק בהתבסס על משוב.**

## (7) פתרון טכנולוגי נבחר

**7.1. טופולוגית הפתרון  
המערכת תתבסס על מבנה שרת-לקוח (Client-Server), שבו החלק הקדמי (Front-end) יטפל בממשק המשתמש והחלק האחורי (Back-end) יטפל בניהול הנתונים, עיבוד לוגיקה ושמירה על אבטחת המידע.**

**7.2 טכנולוגיות בשימוש**

* **צד לקוח (Client) – React, TypeScript.**
* **צד שרת (Server Express) בשילוב Node.ts.**
* **Database - שילוב של MSSQL לנתונים יחסים (כגון משתמשים, קורסים) ו-MongoDB לאחסון בלתי יחסי (כגון חומרים ושיחות).**

**7.3 שפות הפיתוח**

* **Client: TypeScript, JavaScript, HTML, CSS.**
* **Server: TypeScript, JavaScript.**
  1. **) תיאור הארכיטקטורה הנבחרת  
     המערכת תשתמש בארכיטקטורת MVC – Model View Controller לניהול חלקי המידע השונים, לצורך גמישות ופשטות בניהול הקוד.**
  2. **) חלוקה לתכניות ומודולים**
     1. **ממשק משתמש (UI): כולל את כל התצוגות והאינטראקציות מול המשתמש.**
     2. **ניהול נתונים: חלק המערכת המתחבר לבסיסי הנתונים ומבצע שאילתות.**
     3. **ניהול הזדהות (Authentication): יכלול שימוש ב־JWT לצורך אימות משתמש.**
  3. **) סביבת השרת  
     המערכת תתארח בסביבת שרתים מבוססת ענן, כמו Azure או AWS, על מנת להבטיח זמינות גבוהה וביצועים טובים.**

**(7.7) ממשק המשתמש/לקוח – GUI  
עיצוב הממשק יתבסס על עקרונות Material Design כדי לספק חווית משתמש ברורה ומובנת.**

**(7.8) ממשקים למערכות אחרות API -  
המערכת תספק API גמיש שיאפשר אינטגרציה עם מערכות לימוד אחרות ועם מערכות ניהול.**

* 1. **) שימוש בחבילות תוכנה**

**7.9.1. React Router לצורך ניתוב בתוך ה־SPA.**

**7.9.2. Mongoose לניהול חיבור למאגר MongoDB.**

**7.9.3. JWT לצורך הזדהות והצפנת מידע.**

## (8) שימוש במבני נתונים וארגון קבצים

**8.1.**

**A computer screen shot of a computer

Description automatically generated**

**A screenshot of a computer screen

Description automatically generated**

**(8.2) שיטות אחסון**

**שימוש בבסיסי נתונים יחסים (MSSQL) לאחסון מידע מובנה, ובבסיסי נתונים בלתי יחסים (MongoDB) לאחסון קבצים וחומרים.**

**(8.3) מנגנוני התאוששות**

* **גיבויים אוטומטיים: גיבוי יומי של בסיסי הנתונים כדי להבטיח שחזור במקרה של תקלה.**
* **מנגנון ניטור: כל הפעולות נרשמות בלוגים כדי לעקוב אחרי בעיות ולפתור אותן במהירות.**

## (10) תיאור המרכיב האלגוריתמי - חישובי

**10.1. הבעיה שבא לפתור**  
המרכיב האלגוריתמי בפרויקט מתמקד באופטימיזציה של ניהול הזמן והמטלות של הסטודנטים, וכן בהקלה על המרצים בניהול הקורסים והחומרים האקדמיים. הבעיה המרכזית שאנו מנסים לפתור היא חוסר סדר ואי-יכולת לתעדף משימות ומטלות בצורה יעילה, המובילים לאיבוד זמן ולחצים מיותרים מצד הסטודנטים והמרצים כאחד.

**10.2. איסוף מידע וניתוחים סטטיסטיים**

* הישגי קורס מסוים - בדיקה של זמני הגשת מטלות/ פרויקטים לפני מועד ההגשה.
* נוכחות קורסים בזמנים שונים בסמסטר
* ממוצע כיתתי קורסי
* ממוצע תלמידים פרטני
* ממוצע זמן שימוש בטיימר פומודורו בזמן למידה כולל ביום
* הפקת דוחות התקדמות עבור הסטודנטים והמוסדות, על סמך הישגים ותוצאות קודמות.

## (11) אבטחת מידע

**אימות משתמשים (Authentication)**

* המערכת משתמשת ב-JSON Web Token (JWT) לצורך אימות המשתמשים ובקרה על הגישה לנתוני המערכת.
* כל משתמש נדרש להתחבר למערכת באמצעות שם משתמש וסיסמה ייחודיים כדי להבטיח רק גישה מורשית למידע רגיש.

**הרשאות (Authorization)**

* הגבלת גישה לנתונים מתבצעת בהתאם לסוג המשתמש ולתפקידו. לדוגמה, סטודנטים יוכלו לגשת רק לחומרי הלימוד והמטלות הקשורים אליהם, ואילו מרצים יוכלו לצפות ולהתעדכן בנתוני הקורסים שהם מובילים בלבד.

**הצפנת נתונים (Data Encryption)**

* נתונים רגישים במערכת, כמו סיסמאות ואינפורמציה אישית, מוצפנים באמצעות אלגוריתמי הצפנה מתקדמים. זאת כדי להבטיח שאפילו במקרים של דליפת נתונים, המידע יישאר מוגן ולא קריא.

## (12) משאבים הנדרשים לפרויקט

**12.1 . מספר שעות המוקדש לפרויקט וחלוקת עבודה בין חברי הצוות**

הפרויקט דורש כ-500 שעות אדם לפי החלוקה הבאה:

* 50 שעות - אפיון דרישות וניתוח
* 50 שעות - הקמת בסיסי נתונים (MSSQL ו-MongoDB)
* 270 שעות - פיתוח צד לקוח באמצעות React Native
* 130 שעות - פיתוח צד שרת באמצעות Node.js ו-ASP.NET

**12.2. ציוד נדרש**

* מחשבים בעלי ביצועים גבוהים לפיתוח ובדיקות
* חיבור אינטרנט מהיר
* מכשירים ניידים לבדיקת אפליקציות

**12.3. תוכנות נדרשות**

* Visual Studio Code לפיתוח צד לקוח וצד שרת
* Visual Studio לפיתוח צד שרת ב-ASP.NET
* MSSQL Server לניהול בסיס הנתונים
* MongoDB Shell לניהול מסד הנתונים מבוסס NoSQL
* Expo לפיתוח אפליקציות באמצעות React Native

## 12.4. ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט

* פיתוח אפליקציות Native באמצעות React Native
* פיתוח צד שרת באמצעות ASP.NET ו-Node.js
* אינטגרציה בין מסדי נתונים שונים וטכנולוגיות שרת-לקוח
* שימוש ב-JWT לאבטחת מידע ואימות משתמשים

**12.5 ספרות ומקורות מידע**

* [Stack Overflow](https://stackoverflow.com) - תמיכה טכנית
* [W3Schools](https://www.w3schools.com) - חומרי לימוד מקוונים
* [ASP.NET Official Site](https://dotnet.microsoft.com/en-us/apps/aspnet) - תיעוד ASP.NET
* [ChatGPT](https://chatgpt.com) - עזרה בבינה מלאכותית
* [MongoDB](https://www.mongodb.com) - תיעוד MongoDB
* [AI Studio by Google](https://aistudio.google.com) - לימוד ומודלים של AI
* [React Native](https://reactnative.dev) - תיעוד React Native
* [Express.js](https://expressjs.com) - תיעוד Express.js

## (13) תוכנית עבודה ושלבים למימוש הפרויקט

* **שלב 1: אפיון וניתוח דרישות** (שבועות 1-2): הגדרת מטרות הפרויקט וניתוח הדרישות הפונקציונאליות והלא-פונקציונאליות.
* **שלב 2: עיצוב והקמת מסדי נתונים** (שבועות 3-4): עיצוב מסד הנתונים ושילוב ב-MSSQL וב-MongoDB.
* **שלב 3: פיתוח צד לקוח** (שבועות 5-9): בניית הממשק באמצעות React Native ו-Material UI.
* **שלב 4: פיתוח צד שרת** (שבועות 10-13): פיתוח ה-API באמצעות Node.js ו-ASP.NET.
* **שלב 5: בדיקות ואינטגרציה** (שבועות 14-15): ביצוע בדיקות תהליכיות ויחידה, חיבור בין כל המרכיבים ובדיקת תקינות.
* **שלב 6: פריסה וסיום** (שבועות 16-17): השקה והצגת הפרויקט.

## (14) בדיקות

**14.1 בדיקות תהליכיות (Integration Testing)**

* **אינטגרציה בין הלקוח לשרת ב-React ו-ASP.NET Core**:
  + בדיקת תקשורת REST API בין צד הלקוח (React) לצד השרת (ASP.NET Core) כדי לוודא תקינות וקבלת מידע לפי הצורך.
* **אינטגרציה בין השרת ל-SQL Server**:
  + בדיקה שהתקשורת בין השרת לבין מסד הנתונים MSSQL מתבצעת באופן תקין, תוך שליפה, עדכון והוספת נתונים במסד.

**14.2 בדיקות יחידה (Unit Testing)**

* **בדיקת פונקציונאליות של Components ב-React**:
  + כל Component נבדק בנפרד, כולל אופן תצוגת המידע ואינטראקציה עם המשתמש.
* **בדיקת פונקציונאליות של Controllers בשרת ASP.NET Core**:
  + כל Controller נבדק בנפרד כדי לוודא שהוא מגיב נכון לבקשות ומספק את התוצאות הנדרשות בהתאם לדרישות הפונקציונאליות של המערכת.

## תוכנית בדיקות תוכנה (STP)

**שם התוכנית**: התחברות

**מתכנן הבדיקה**: מיכאל קולומיזה

**תאריך**: 10/10/2024

**גרסה**: 0.0.1

**מבוא**

תוכנית בדיקה זו נועדה לבדיקת הפונקציונאליות של ההתחברות לאתר AcademiQ. מטרת הבדיקה היא לוודא שניתן להתחבר לאתר לאחר מילוי כל הנתונים הנדרשים ולהבטיח שכל השדות וההגבלות פועלים כראוי.

**היקף הבדיקה**

בדיקות אלו יכללו:

* בדיקות פונקציונאליות של כל שדה בטופס ההרשמה.
* בדיקות ולידציה של שדות הכניסה (כגון ת"ז, סיסמא').
* בדיקות התנהגות המערכת בעת מילוי שגוי של שדות.
* בדיקות תהליך ההתחזרות עד מעבר לדף הראשי בהצלחה.

**אסטרטגיית בדיקה**

**סוגי הבדיקות**

* בדיקות פונקציונאליות: בדיקת ההתנהגות של המערכת על פי הדרישות.
* בדיקות אבטחה: בדיקת האבטחה של המערכת (לדוגמה, אימות משתמשים, הגנה על סיסמאות).
* בדיקות UI: בדיקת ממשק המשתמש (לדוגמה, בדיקת העיצוב, הנוחות והאינטואיטיביות של הממשק).

**כלים לבדיקות**

* כלי בדיקות ידניות (כגון TestRail).

**משאבים ולוח זמנים**

**משאבים**

* **צוות בדיקות**: חברי צוות הפיתוח.
* **כלי בדיקה**: מחשבים עם גישה אינטרנט.

**תנאי קבלה וקריטריונים להשלמה**

**תנאי קבלה**

* API של המערכת זמינים ונגישים.
* כל הבדיקות הפונקציונאליות עברו בהצלחה.
* כל הבדיקות ולידציה עברו בהצלחה.
* כל מקרי הקצה נבדקו והמערכת התנהגה כמצופה.
* אין בעיות קריטיות פתוחות.

**קריטריונים להשלמה**

* סיום כל תרחישי הבדיקה שתוכננו.
* פתרון כל הבאגים הקריטיים.
* המערכת צריכה לעמוד בדרישות של המפרט הטכני.

**ניהול סיכונים**

**סיכונים פוטנציאליים**

* בעיות אבטחה.
* קושי בתאימות לדרישות המערכת.
* זמן בדיקה ארוך מהמתוכנן עקב בעיות לא צפויות.
* בעיות טכניות בכלי הבדיקות.
* באגים קריטיים שמתגלים בשלבים מאוחרים.

**תוכנית ניהול סיכונים**

* ביצוע הערכת סיכונים לפני תחילת הבדיקות.
* מעקב שוטף אחר זמן הבדיקות ומצב הבאגים.

**נספחים**

* **תסריטי בדיקה מפורטים**: מצורפים במסמך נפרד.
* **דוחות בדיקה**: דוחות מתועדים לאחר כל הרצת בדיקה.

**שם הפרויקט**: AcademiQ

**שם התוכנית**: שכחתי סיסמא

**מתכנן הבדיקה**: מיכאל קולומיזה

**תאריך**: 12/10/2024

**גרסה**: 0.0.1

**מבוא**

תוכנית בדיקה זו נועדה לבדיקת הפונקציונאליות של מנגנון שכחת סיסמא באתר AcademiQ. מטרת הבדיקה היא לוודא שניתן לשחזר משתמש אם אין סיסמא ולהבטיח שכל השדות וההגבלות פועלים כראוי.

**היקף הבדיקה**

בדיקות אלו יכללו:

* בדיקות פונקציונאליות של כל שדה בטופס "שכחתי סיסמא".
* בדיקות ולידציה של שדות הכניסה (כגון אימייל, ת"ז).
* בדיקות התנהגות המערכת בעת מילוי שגוי של שדות.
* בדיקות תהליך שחזור הסיסמא עד לקבלת הודעת שחזור סיסמא ויצירת סיסמא חדשה.

**אסטרטגיית בדיקה**

**סוגי הבדיקות**

* בדיקות פונקציונאליות: בדיקת ההתנהגות של המערכת על פי הדרישות.
* בדיקות אבטחה: בדיקת האבטחה של המערכת (לדוגמה, אימות משתמשים, הגנה על סיסמאות).
* בדיקות UI: בדיקת ממשק המשתמש (לדוגמה, בדיקת העיצוב, הנוחות והאינטואיטיביות של הממשק).

**כלים לבדיקות**

* כלי בדיקות ידניות (כגון TestRail).

**משאבים ולוח זמנים**

**משאבים**

* **צוות בדיקות**: חברי צוות הפיתוח.
* **כלי בדיקה**: מחשבים עם גישה אינטרנט.

**תנאי קבלה וקריטריונים להשלמה**

**תנאי קבלה**

* API של המערכת זמינים ונגישים.
* כל הבדיקות הפונקציונאליות עברו בהצלחה.
* כל הבדיקות ולידציה עברו בהצלחה.
* כל מקרי הקצה נבדקו והמערכת התנהגה כמצופה.
* אין בעיות קריטיות פתוחות.

**קריטריונים להשלמה**

* סיום כל תרחישי הבדיקה שתוכננו.
* פתרון כל הבאגים הקריטיים.
* המערכת צריכה לעמוד בדרישות של המפרט הטכני.

**ניהול סיכונים**

**סיכונים פוטנציאליים**

* בעיות אבטחה.
* קושי בתאימות לדרישות המערכת.
* זמן בדיקה ארוך מהמתוכנן עקב בעיות לא צפויות.
* בעיות טכניות בכלי הבדיקות.
* באגים קריטיים שמתגלים בשלבים מאוחרים.

**תוכנית ניהול סיכונים**

* ביצוע הערכת סיכונים לפני תחילת הבדיקות.
* מעקב שוטף אחר זמן הבדיקות ומצב הבאגים.

**נספחים**

* **תסריטי בדיקה מפורטים**: מצורפים במסמך נפרד.
* **דוחות בדיקה**: דוחות מתועדים לאחר כל הרצת בדיקה.

## תוכנית בדיקות תוכנה (STD)

**שם התסריט:** התחברות

**שם מתכנן הבדיקה:** מיכאל קולומיזה **תאריך:** 12/10/2022 **סעיף באפיון:** 1

**רמת הבדיקה:** פונקציונאלית **עדיפות:** גבוהה **סטטוס: בתהליך**

**מטרה:**

מטרת הבדיקה היא לוודא שיש אפשרות להתחבר למערכת לחשבון של משתמש

**דרישות קדם:** ת"ז וססמא של משתמש רשום.

**ניווט למסך (אם יש):**

יש להיכנס AcademiQ ולאחר מיכן על כפתור ההתחברות.

**צעדי הבדיקה:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **מס"ד** | **שם הבדיקה** | **תיאור הבדיקה** | **תוצאה רצויה** | **תוצאה מתקבלת** |
| 1 | פוקוס על תיבת "תז" | לחיצה על תיבת הטקסט | הכותרת תעלה כך שתהיה מוצגת | V |
| 3 | פוקוס על תיבת "סיסמא" | לחיצה על תיבת הטקסט | הכותרת תעלה כך שתהיה מוצגת | **V** |
| 4 | כתיבה בתיבת "תז" | לכתוב בתיבה.  שתהיה אפשרות להקיש רק מספרים | שתיהיה אפשרות להקיש רק מספרים, לא יופיעו אותיות | **V** |
| 5 | כתיבה בתיבת "סיסמא" | לכתוב בתיבה ססמא | הסיסמא תהיה חסויה | **V** |
| 6 | ההתחברות | לחיצה על כפתור ההתחברות אם נתונים נכונים | מעבר לעמוד הבית | **V** |
| 7 | לחיצה על כפתור ההתחברות אם נתונים שגואים | הודעה המודיעה על נתונים שגואים | **V** |

**שם התסריט:** שכחתי סיסמא

**שם מתכנן הבדיקה:** מיכאל קולומיזה **תאריך:** 12/10/2022 **סעיף באפיון:** 2

**רמת הבדיקה:** פונקציונאלית **עדיפות:** גבוהה **סטטוס: בתהליך**

**מטרה:**

מטרת הבדיקה היא לוודא שמנגנון שחזור הסיסמא עובד

**דרישות קדם:** ת"ז ומייל של משתמש רשום

**ניווט למסך (אם יש):**

יש להיכנס לAcademiQ ללחוץ על כפתור ההתחברות ואז על כפתור "שכחתי סיסמא".

**צעדי הבדיקה:**

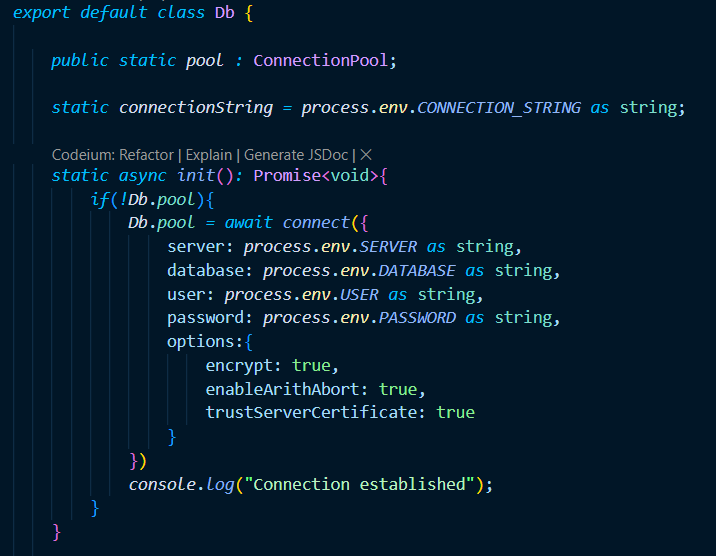
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **מס"ד** | **שם הבדיקה** | **תיאור הבדיקה** | **תוצאה רצויה** | **תוצאה מתקבלת** |
| 1 | פוקוס על תיבת "תז" | לחיצה על תיבת הטקסט | הכותרת תעלה כך שתהיה מוצגת | V |
| 3 | פוקוס על תיבת "מייל" | לחיצה על תיבת הטקסט | הכותרת תעלה כך שתהיה מוצגת | **V** |
| 4 | כתיבה בתיבת "תז" | לכתוב בתיבה.  שתהיה אפשרות להקיש רק מספרים | שתיהיה אפשרות להקיש רק מספרים, לא יופיעו אותיות | **V** |
| 5 | כתיבה בתיבת "מייל" | לכתוב בתיבה מייל.  שתהיה אפשרות להקיש רק אותיות באנגלית וסמנים חוקיים למייל | לא יופיעו אותיות שלא באנגלית וסמנים לא חוקיים במייל | **V** |
| 6 | שליחת הנתונים | לחיצה על כפתור שליחת הנתונים, אם נתונים נכונים | מעבר לעמוד הבית וקבלת מייל עם קישור | **V** |
| 7 | לחיצה על כפתור שליחת הנתונים, אם נתונים שגואים | הודעה המודיעה על נתונים שגואים | **V** |
| 8 | כניסה לקישור | בדיקה אם מופיעה במייל קישור לשיחזור סיסמא | קישור במייל | **V** |
| 9 | לחיצה על הקישור | מעבר לעמוד חידוש הסיסמא | **V** |
| 10 | שיחזור סיסמא | הקלדת סיסמא חדשה | סיסמא מוסתרת | **V** |
| 11 | הקלדת סיסמא פעם נוספת | סיסמא מוסתרת | **V** |
| 12 | הקלדת סיסמא לא תואמת | הודעה המודיעה על נתונים שגואים | **V** |

**טבלת פירוט מקרים**

**סיכום ( + הצעות לשיפור):**

**הוספת כפתור להצגת הסיסמא.**

## דוגמאות קוד



A computer screen shot of code

Description automatically generated

A screen shot of a computer code

Description automatically generated

A computer screen shot of text

Description automatically generated