

מסמך ייזום

UniSmart

שמות חברי הצוות:

אביטל זר - 331659516

מרים נגר - 213856198

אסתי למקין - 212128490

למה בחרנו בפרויקט:

הפרויקט אמור לעזור לסדר מערכת שעות לסטודנטים בקלות.

במהלך הלימודים באוניברסיטה נתקלנו לא פעם ולא פעמיים בקשיים בביצוע המשימה המסובכת: סידור מערכת שעות; התאמה בין קורסים, בחירה בין מועד אחד למשנהו והתאמה לשעות העבודה והחובות האישיות.

עקב כך החלטנו לבחור אפליקציה לפרויקט שתקל את המשימה באמצעות בניית אלגוריתמים שיעזרו בחיפוש אחר מערכת שעות מתאימה לכל סטודנט. אפליקציה שתעזור לנו עצמנו, ואולי תוכל לעזור לאלפי סטודנטים ולהקל עליהם בבניית המערכת.

חלק 1 - מקרה עסקי ומסמך ייזום

שם פרויקט: UniSmart

ציון הבעיה:

סטודנטים באוניברסיטאות רבות נתקלים בקושי משמעותי בבניית מערכת שעות יעילה לסמסטר. תהליך הבחירה בקורסים מבוסס לרוב על ניסוי וטעייה: כל סטודנט נדרש לבדוק ידנית אילו קורסים זמינים, אילו חוסמים זה את זה, אילו מרצים מלמדים, ומהם החלונות הפנויים בימים ובשעות הרצאות. בנוסף, אין כיום כלי מרכזי שמאפשר התאמה אישית חכמה של מערכת השעות על סמך העדפות הסטודנט (כמו ימי לימוד מועדפים, זמינות בתחבורה ציבורית, קורסים עם חברים, או מרצים מועדפים). בעיות נוספות כוללות:

זמינות מקומות פנויים משתנה בקורסים ללא התרעה בזמן אמת במידה ואתה לא מחובר למערכת המודל. קושי לזהות חלופות מתאימות במקרה שקורס מסוים נסגר. בזבוז זמן רב על ניסיונות חוזרים ונשנים לשבץ מערכת שמתאימה לכל המגבלות. לכן נדרשת מערכת שתאפשר גנרציה אוטומטית של מערכות שעות חכמות בהתאם למאפיינים האישיים של כל סטודנט, ותתריע על שינויים רלוונטיים בזמן אמת.

Market Research – סקירת פתרונות קיימים:

UniTime/TimetableMaster - פתרון עבור בניית מערכת שעות של מוסדות, כגון האוניברסיטה עצמה, שלא מיועד לסטודנט פרטי.

Coursicle — אפליקציה לבניית מערכת שעות לסטודנטים בארה"ב שמותאמת ללוח הזמנים של כל אוניברסיטה. לא מותאמת להעדפות אישיות של סטודנטים, ולא מותאמת לאוניברסיטאות בישראל.

יישומי תזמון/Planner אישיים (MyStudyLife, Smart Timetable ועוד) — אפליקציות כלליות לניהול זמן שלא קשורות ישירות לבניית מערכת, בפרט לא של אוניברסיטה.

Bid it — אפליקציה לשימוש של סטודנטים באוניברסיטת תל אביב, שנותנת סידור של מערכת על פי רשימה של קורסים ויש אופציה להוספת העדפות סטודנט אבל מינימלית.

מטרות:

פרויקט UniSmart נועד לתת מענה לצורך ממשי בקרב סטודנטים בבניית מערכת שעות אקדמית יעילה ונוחה לשימוש.

תהליך בניית מערכת שעות כיום הוא ארוך, מסורבל ולעיתים מתסכל – הסטודנט נדרש לבדוק ידנית אילו קורסים מתנגשים, אילו הם חובה או בחירה, ולנסות להתאים בין עשרות אפשרויות שונות, תוך התחשבות בהעדפות אישיות כגון ימים פנויים, שעות מועדפות או מרצים מסוימים.

המטרה שלנו היא שעל סמך הקורסים שהסטודנט רוצה להירשם והמגבלות זמן שלו, האפליקציה תדע לסדר את מערכת השעות האופטימלית עבורו, ולהציע מספר אפשרויות למערכת ותקל במידה משמעותית את כמות הזמן שלוקח לבנות את המערכת ואת הסיבוך שבבנייה.

האפליקציה תהווה תוספת חיצונית לאוניברסיטאות שונות שבחיבור למסד נתונים של הקורסים יהיה ניתן לגנוב את מערכת השעות.

יעדים:

היעדים המרכזיים של הפרויקט כוללים:

- קליטת נתוני משתמש - הסטודנט יוכל להזין רשימה של קורסים מבוקשים לסמסטר הקרוב.
- הזנת מגבלות והעדפות - אפשרות להגדיר ימים פנויים, שעות מועדפות, מרצים מועדפים או קורסים שיש להימנע מהם.
- גנרציה חכמה של מערכת שעות - המערכת תיצור באופן אוטומטי מספר אפשרויות שונות למערכת שעות אופטימלית, תוך התחשבות בחסימות בין קורסים ובקריטריונים שהוזנו.
- ממשק גרפי אינטואיטיבי - הצגה ויזואלית ברורה של מערכת השעות, כולל תצוגת השוואה בין כמה חלופות.
- חיבור למסד נתונים אוניברסיטאי - אינטגרציה עם מאגר הקורסים של האוניברסיטה (באמצעות API או קובצי נתונים), לצורך משיכת נתוני קורסים בזמן אמת.
- התראות בזמן אמת - המערכת תתריע למשתמש כאשר מתפנה מקום בקורס, מתווספת קבוצה חדשה או נפתחת אפשרות להירשם לקורס בחירה נוסף.
- מערכת ניהול (Admin Dashboard) - עבור הנהלת האוניברסיטה – תצוגה של סטטיסטיקות רישום, מספר נרשמים לכל קורס, וביקוש לקורסי בחירה.

יתרונות:

- חיסכון משמעותי בזמן
- רישום מהיר יותר ולכן עומס פחות על מערכת האוניברסיטה
- מערכת מדויקת יותר לצרכי הסטודנט מה שמאפשר גמישות וסדר ביומיום ובחיים הפרטיים
- מאפשרת חיבור למידע אקדמי אמיתי (מערכי קורסים, מרצים, כיתות).

השפעה צפויה:

על הסטודנטים – האפליקציה תקל משמעותית על תכנון המערכת ותאפשר בחירה מושכלת ונוחה של קורסים, תוך הפחתת עומס ולחץ בתקופת ההרשמה. שיפור חוויית המשתמש יגביר את שביעות הרצון של הסטודנטים מהמערכת האקדמית.

על האוניברסיטה – תצמצם פניות ותמיכה טכנית מצד הסטודנטים ותייעל את תהליך ניהול הקורסים.

היקף פרויקט:

פרויקט UniSmart כולל פיתוח אפליקציה חכמה לבניית מערכת שעות אקדמית מותאמת אישית לסטודנטים.

האפליקציה תספק ממשק משתמש אינטואיטיבי וידידותי, אשר יאפשר לסטודנט להזין את רשימת הקורסים הרצויים, את המגבלות האישיות שלו (ימים פנויים, שעות מועדפות, מרצים מועדפים וכו'), ולקבל באופן אוטומטי מספר הצעות למערכות שעות אופטימליות.

בנוסף, המערכת תתממשק למסד נתונים אוניברסיטאי או API חיצוני לצורך משיכת נתוני קורסים בזמן אמת, ותאפשר שליחת התרעות על שינויים בזמינות קורסים.

עבור גורמי ניהול (מזכירות או מנהלי מערכת), תינתן אפשרות צפייה בסטטיסטיקות רישום ובנתוני עומס בקורסים.

נכלל:

- פיתוח אפליקציה למוביל עם ממשק משתמש נוח ואינטואיטיבי.
- פיתוח מנוע גנרציה חכם למערכות שעות המבוסס על אילוצים והעדפות.
- יצירת חיבור למסד נתונים אוניברסיטאי או API קיים של נתוני קורסים.
- שליחת התרעות בזמן אמת (push notifications) על פתיחת מקומות או שינויים במערכת הקורסים.
- יצירת ממשק ניהול בסיסי להצגת נתונים וסטטיסטיקות.

- בדיקות קצה ושימושיות להבטחת יציבות וחויית משתמש חיובית.

לא נכלל:

האפליקציה אינה מיועדת לביצוע רישום בפועל לקורסים במערכת האוניברסיטה (אלא רק לסייע בבנייה ובהתאמה של המערכת).

בשלב הראשוני לא תתבצע אינטגרציה מלאה עם מערכות לו"ז אקדמי של אוניברסיטאות למעט אריאל

האלגוריתם יתמקד באופטימיזציה לפי מערכת העדפות קבועה מראש

ממשק הניהול הראשוני יכלול רק טבלאות בסיסיות של נתונים ולא ניתוחים מתקדמים של הסקת מסקנות מתוך הנתונים

סיכונים עיקריים:

- חוסר גישה למידע עדכני על זמני קורסים באוניברסיטת אריאל
השפעה אפשרית: עיכוב בשלבים הראשונים של הפיתוח או קושי בבדיקת האלגוריתם על נתונים אמיתיים.
ניהול סיכון: ניסיון לקבל גישה למסד הנתונים בשלב מוקדם של הפרויקט. במידה ולא תתאפשר גישה – ייבנה אב-טיפוס המבוסס על נתוני דמה, שישימש להוכחת היתכנות (Proof of Concept) ויוכל לשמש כתוסף פוטנציאלי לאוניברסיטה בהמשך.
- קושי בהתאמת האפליקציה למוסדות אקדמיים שונים
השפעה אפשרית: הגבלה של השימוש לאוניברסיטת אריאל בהתחלה.
ניהול סיכון: בשלב הראשון יתבצע מיקוד בפיתוח והתאמה לאוניברסיטת אריאל בלבד. לאחר הצלחת האבטיפוס ניתן יהיה להרחיב את התמיכה באמצעות מבנה מודולרי המאפשר התאמה למסדי נתונים שונים.
- אי מציאת מערכת שעות העומדת בכל המגבלות שהוזנו על ידי הסטודנט
השפעה אפשרית: המערכת לא תממש את המטרה שלה במקרים מסובכים.
ניהול סיכון: הצגת מערכת קרובה לאופטימלית יחד עם הסבר ברור אילו אילוצים מנעו מציאת פתרון מלא, כך שהסטודנט יוכל לבחור אילוצים להקלה או לשינוי.
- מורכבות גבוהה באלגוריתמי האופטימיזציה
השפעה אפשרית: עומס חישובי גבוה או זמני ריצה ארוכים מדי.
ניהול סיכון: בחינת מספר אלגוריתמים שונים ובחירה באלגוריתם היעיל ביותר לאחר התייעצות עם צוות הקורס וסקירת פתרונות קיימים בספרות או בקוד פתוח.

חלופות שנשקלו

- בניית תוסף (Plugin) ישירות לאתרי האוניברסיטאות
חלופה זו הייתה טבעית מכיוון שהמידע הנדרש קיים כבר באתרי הרישום האקדמיים.
עם זאת, היא נדחתה ממספר סיבות:
 1. הדרישה בקורס היא לפתח אפליקציית מובייל מלאה – ולכן תוסף אתר אינו עומד בדרישות.
 2. היעדר גישה לקוד המקור של אתרי האוניברסיטאות מקשה על שילוב ישיר.
 3. מגבלות אבטחה, API סגור ודרישות הרשאה מורכבות היו הופכים את הפיתוח לאיטי ומסובך יותר.
- אפליקציה המבוססת על הזנת נתונים ידנית של הקורסים
נשקלה האפשרות לפתח אפליקציה שבה הסטודנט מזין בעצמו את כל זמני הקורסים והתרגולים.
חלופה זו נדחתה מהסיבות הבאות:
 1. תהליך הזנה ידנית מתיש, ארוך ומועד לטעויות.
 2. החוויה הייתה פחות חדשנית ופחות יעילה, בניגוד לדרישת הקורס לחדשנות.
 3. אפליקציה שמסתמכת על הזנות ידניות אינה נותנת ערך מוסף משמעותי לעומת פתרונות קיימים.
- בניית מערכת שולחנית במקום אפליקציית מובייל
האפשרות לשימוש בפלטפורמה שולחנית נדחתה משום:
 1. דרישת הקורס היא אפליקציית מובייל שניתנת להפצה בחנויות האפליקציות.
 2. רוב הסטודנטים משתמשים במכשירים ניידים בזמן רישום לקורסים, ולכן אפליקציה ניידת רלוונטית ונגישה יותר.

מדוע הפתרון הנוכחי הוא המועדף:

- הבחירה באפליקציית מובייל ליצירת מערכת שעות חכמה היא הפתרון האופטימלי מכמה סיבות מרכזיות:
1. התאמה מלאה לדרישות הקורס – אפליקציית מובייל, עם Authentication, Notifications, עבודה מול Database ושימוש בסנסורים.
 2. ערך גבוה לסטודנטים – חיסכון משמעותי בזמן, הפחתת לחץ בתקופת רישום והצגת מערכות שעות מותאמות אישית.
 3. חדשנות – שילוב אוטומטי של מגבלות אישיות
 4. גמישות ויכולת הרחבה – המערכת יכולה לתמוך ב-API של אוניברסיטאות שונות או לעבוד עם נתוני דמה.
 5. ממשק משתמש נגיש ומותאם לדור הנייד – שימושיות גבוהה יותר מאתרי אוניברסיטה מסורבלים.

בעלי עניין עיקריים:

- המרצה והמנחים בקורס הנדסת תוכנה באוניברסיטת אריאל.
- סיבה: אחראים על אישור הפרויקט, ליווי מקצועי והערכת תוצרי הפיתוח.
- צוות הפיתוח - צוות הפרויקט
- סיבה: אחראים על אפיון, תכנון, פיתוח, בדיקות והצגת האפליקציה.
- משתמשי קצה - סטודנטים באוניברסיטאות בישראל, בראש ובראשונה באוניברסיטת אריאל.
- סיבה: המשתמשים העיקריים באפליקציה לבניית מערכת שעות חכמה ונוחה.
- מוסדות לימוד המעוניינים לשלב את המערכת בעתיד.
- סיבה: מגישים את האפליקציה לסטודנטים.
- בעלי עניין משני – משתמשים בממשק הניהול לצפייה בסטטיסטיקות רישום ובנתוני ביקוש לקורסים.
- מתרגלי הקורס או מומחים באלגוריתמים ואופטימיזציה, אשר יספקו ייעוץ טכני בבחירת שיטות הפתרון ובבקרת איכות הקוד.

משאבים נדרשים

- צוות פיתוח.
- מחשבים, טלפונים.
- סביבת פיתוח - Visual Studio Code, android studio, firebase, github, git.
- מסד עם נתונים על הקורסים של האוניברסיטה.
- כלי לניהול משימות - trello.

חלק 2 - הצהרת עבודה

הצהרת חזון:

חזון פרויקט UniSmart הוא ליצור מערכת חכמה, אינטואיטיבית ומבוססת נתונים שתשנה את הדרך בה סטודנטים בונים את מערכת השעות האקדמית שלהם.

האפליקציה תשמש עזר דיגיטלי לתכנון מערכת שעות, המסוגל להבין את העדפות הסטודנט, לנתח אילוצים וזמני קורסים, ולהציע בזמן אמת פתרונות מותאמים אישית — במהירות, בדיוק ובנוחות מרבית.

מטרת הפרויקט היא לספק חוויה אחידה, פשוטה ונגישה לכל סטודנט, ללא צורך בבדיקות ידניות או בחיפוש מתיש של שילובים אפשריים.

בנוסף, המערכת תתממשק למידע אוניברסיטאי אמיתי ותספק ערך גם למוסדות עצמם, באמצעות תובנות וסטטיסטיקות על דפוסי רישום וביקוש לקורסים.

יעדים מרכזיים של החזון:

לאפשר לכל סטודנט לבנות מערכת שעות מלאה בתוך דקות, בהתאמה אישית למגבלותיו והעדפותיו.

לשפר את חוויית השימוש במערכות אקדמיות באמצעות ממשק מודרני וקל לשימוש.

לספק מערכת מודולרית שניתנת להרחבה לאוניברסיטאות נוספות ולממשקים קיימים.

לתרום להפחתת העומס על מערכות הרישום וכך לייעל את תהליך תכנון הסמסטר ברמה מוסדית.

היקף פרויקט:

פרויקט UniSmart מתמקד בפיתוח אפליקציה חכמה לניהול ובניית מערכות שעות אקדמיות מותאמות אישית לסטודנטים.

הגבולות של הפרויקט כוללים:

Within Scope (בתחום הפרויקט):

- מערכת Authentication הכוללת הרשמה עם שם משתמש וסיסמה, וכן התחברות דרך Google ו-Facebook.
- פיתוח אפליקציה למובייל (Android/iOS) עם מנוע גנרציה חכם למערכות שעות.
- אינטגרציה עם מסד נתונים אוניברסיטאי או API לקבלת מידע על קורסים, מרצים וזמני שיעור.
- יצירת ממשק משתמש אינטואיטיבי עם תפריט ניווט אינטראקטיבי והצגת מערכות שעות והעדפות.
- שליחת התרעות על שינויים במערכת הקורסים (כגון פתיחת מקומות).
- פיתוח ממשק ניהול בסיסי להצגת סטטיסטיקות רישום לקורסים.

- תמיכה בשני סוגי משתמשים לפחות:
- משתמש רגיל (סטודנט) – יכול להזין קורסים, לקבל הצעות למערכות שעות, ולקבל התראות.
- משתמש מנהל – יכול לצפות באינפורמציה של רישום, לנהל מידע ולהציג נתונים סטטיסטיים.
- Out of Scope (מחוץ לתחום הפרויקט):
- ביצוע רישום אוטומטי או הרשמה לקורסים במערכות האוניברסיטה.
- אינטגרציה מלאה עם מערכות רישום אוניברסיטאיות סגורות שאינן מאפשרות גישה ל-API.
- פיתוח תכונות מורכבות כגון חישוב עומס נק"ז, ניתוח עומק של נתוני רישום, או התאמה אישית מורכבת של סגנונות הוראה.

היקף העבודה:

העבודה בפרויקט מחולקת למספר שלבים מרכזיים הכוללים:

- אפיון ותכנון מערכת:
 - הגדרת דרישות המשתמש והמערכת.
 - תכנון ארכיטקטורת התוכנה ומבנה בסיס הנתונים.
 - עיצוב ממשק המשתמש.
- פיתוח:
 - יישום מערכת Authentication עם אפשרות כניסה דרך גוגל ופייסבוק.
 - פיתוח מנוע הגרציה החכם לבניית מערכות שעות בהתחשב במגבלות.
 - פיתוח ממשק המשתמש לאפליקציה למובייל.
 - יצירת חיבור למסד הנתונים או API חיצוני לקבלת נתוני קורסים.
 - פיתוח מנגנון התראות בזמן אמת.
 - אינטגרציה עם חיישני הטלפון.
 - פיתוח תפריט ניווט ועבודה על UX נוח.
- בדיקות:
 - בדיקות פונקציונליות של כל רכיבי המערכת.
 - בדיקות שימושיות וחווית משתמש.
 - תיקון באגים ושיפור ביצועים.
- הטמעה והפצה:
 - הכנת גרסת הפצה ראשונית (Release Candidate).
 - הדרכה ראשונית למשתמשים (סטודנטים ומנהלים).

תכונות מרכזיות:

- הזנת רשימת קורסים שהסטודנט רוצה ללמוד לסמסטר הקרוב.
- הזנת מגבלות אישיות כגון ימים פנויים, שעות מועדפות, שעות עבודה ומרצים מועדפים.

- גנרציה אוטומטית של מספר אפשרויות למערכת שעות אופטימלית בהתחשב במגבלות וקורסים חוסמים.
- ממשק משתמש ויזואלי אינטואיטיבי להצגת מערכות השעות והאפשרויות השונות.
- חיבור למסד נתונים אוניברסיטאי או API לקבלת נתוני קורסים בזמן אמת.
- מערכת התראות בזמן אמת (Push Notifications) על שינויים במצב הקורסים, כגון פתיחת מקומות.
- ממשק ניהול בסיסי להצגת סטטיסטיקות רישום וביקוש לקורסים עבור צוותי ניהול אוניברסיטאיים.

מגבלות

- מגבלת זמן פיתוח בהתאם למסגרת הקורס (כשלושה חודשים).
- הגבלה טכנולוגית בהתאם לסביבת הפיתוח שנבחרה - android studio מיועד לפיתוח עבור אנדרואיד בלבד.
- גישה מוגבלת למסדי נתונים של האוניברסיטה, שתתאפשר רק באמצעות API פתוח או נתוני דמה במידת הצורך.
- תקציב מוגבל ולכן יש להסתמך על כלים וטכנולוגיות חינוכיות עבור כלי הפיתוח וכלי ניהול המשימות. שימוש בטכנולוגיות קוד פתוח בלבד.

תלויות

- זמינות וגישה למסד הנתונים או API של האוניברסיטה לקבלת מידע עדכני על הקורסים, המרצים וזמני השיעורים.
- תלות בטכנולוגיות צד שלישי לפיתוח האפליקציה (כגון Android studio, Firebase ועוד).
- תלות בסביבת פיתוח וכלי ניהול גרסאות (GitHub/GitLab) לצורך עבודה צוותית.
- תלות במשתמשי הקצה (סטודנטים ואנשי ניהול) לשם בדיקות שימושיות ואיסוף משוב.
- זמינות ותמיכה של צוות הקורס ומנחים מקצועיים להכוונה וייעוץ טכני לאורך הפרויקט.
- זמינות צוות הפיתוח: תלות בזמינות חברי הצוות במסגרת סמסטר עם מטלות ולימודים נוספים.

תוצרים

- אפליקציה חכמה למובייל – המאפשרת לסטודנטים להזין קורסים ומגבלות ולקבל הצעות למערכות שעות מותאמות אישית.
- מנוע גנרציה חכם – אלגוריתם אופטימיזציה המייצר מערכות שעות בהתאם לקריטריונים והאילוצים שהוזנו.
- ממשק משתמש אינטואיטיבי – עיצוב ויישום ממשק משתמש נוח להצגת מערכות השעות והאפשרויות השונות.
- חיבור למסד נתונים / API אוניברסיטאי – ממשק תקשורת עם מסד נתונים לקבלת מידע עדכני על קורסים, מרצים וזמני שיעור.

- מערכת התרעות בזמן אמת (Push Notifications) – התראות על שינויים במצב הקורסים, כגון פתיחת מקומות פנויים.
- ממשק ניהול בסיסי (Admin Dashboard) – הצגת סטטיסטיקות רישום וביקוש לקורסים לצוותי ניהול במוסדות האקדמיים.
- תיעוד טכני ומדריך משתמש – מסמכי אפיון, תיעוד קוד, ומדריך משתמש פשוט וברור להפעלת האפליקציה.
- דו"ח בדיקות – תיעוד תוצאות הבדיקות הפונקציונליות, שימושיות וביצועים שבוצעו במהלך הפיתוח.
- מסמכי הליך הפרויקט: מסמך יזום, מסמך דרישות, מסמך ניתוח.
- ניהול הפרויקט באמצעות Trello - לוח משימות מעודכן עם פירוט השלבים וההתקדמות לאורך כל הפרויקט.

לוח זמנים

1. יצירת קשר עם האוניברסיטה והכנת נתונים
 - יצירת קשר עם גורמי האוניברסיטה לקבלת גישה למסד הנתונים או API.
 - איסוף מידע ראשוני והכנת נתוני דמה במידת הצורך.
 - משך משוער: 1–2 שבועות.
 - אבני דרך: אישור גישה למסד הנתונים או פתרון חלופי.
2. אפיון ופיתוח ראשוני (כולל Authentication)
 - אפיון דרישות בסיסיות במקביל לפיתוח פונקציות Authentication.
 - פיתוח ממשק משתמש ראשוני והתקנת סביבת פיתוח.
 - משך משוער: 3–4 שבועות.
 - אבני דרך: אב-טיפוס ראשוני הכולל Authentication ומסד נתונים.
3. פיתוח מתקדם
 - יישום מנוע גנרציה למערכות שעות.
 - אינטגרציה עם חיישני הטלפון
 - מימוש התראות Push Notifications.
 - משך משוער: 3 שבועות.
 - אבני דרך: גרסת בטא פנימית.
4. בדיקות ואופטימיזציה
 - ביצוע בדיקות פונקציונליות ושימושיות.
 - תיקון באגים ושיפור ביצועים.
 - משך משוער: 2 שבועות.
 - אבני דרך: אישור גרסת Release Candidate.

5. פריסה ותיעוד

- הכנת תיעוד משתמש ומפתח.
- מסירת הפרויקט.
- משך משוער: 1 שבוע.
- אבני דרך: הפרויקט מוגש ומוכן להפצה.

קריטריוני קבלה

הקריטריונים להערכת הצלחת הפרויקט וקבלת תוצר העבודה כוללים:

תפקוד מלא של הפיצ'רים המרכזיים-

- מערכת Authentication עובדת בצורה תקינה עם התחברות דרך שם משתמש וסיסמה, וכן דרך Google ו-Facebook.
- יצירת מערכות שעות חכמות על פי המגבלות שהוזנו ע"י המשתמש, עם הצגת מספר אפשרויות.
- קבלת התראות בזמן אמת (Push Notifications) על שינויים במערכת הקורסים.
- אינטגרציה עם שני חיישני טלפון.
- שימוש במסד נתונים שבו יש את נתוני הקורסים.
- תמיכה במשתמש רגיל - סטודנט, ומשתמש מנהל.

יציבות וביצועים-

- זמני תגובה של האפליקציה עד 5 דק בפעולות בפעולת גנרוט המערכת.
- האפליקציה לא תקרוס במהלך שימוש שגרתי, ותתמודד עם טעויות בצורה מאוזנת.

חויית משתמש-

- תפריט - ניווט קל וברור בין המסכים ובין הפיצ'רים השונים.

עמידה בלוח הזמנים-

- סיום כל שלבי הפיתוח והבדיקות בהתאם ל-Timeline שהוגדר.
- שימוש בכלים ובמשאבים במסגרת המגבלות שהוגדרו.
- שיקוף ניהול המשימות לאורך הפרויקט

ניהול סיכונים

זיהוי הסיכונים:

1. גישה מוגבלת למסדי נתונים של האוניברסיטה
קושי בקבלת גישה בזמן או באיכות מידע לא מספקת על זמני קורסים ומרצים, דבר שעלול לפגוע ביכולת האפליקציה להציע מערכות שעות מדויקות.

2. מחסור בזמן ובמשאבים
מגבלת זמן הפרויקט (כשלושה חודשים) ותחומי ידע טכנולוגיים עשויים לגרום לעיכובים או לפשרות באיכות המוצר הסופי.

אסטרטגיות התמודדות:

- לגבי גישה למסדי נתונים: יש להתחיל את תהליך יצירת הקשר עם האוניברסיטה כבר בשלב מוקדם, ולהיערך מראש לעבודה עם נתוני דמה במקרה הצורך.
- לגבי משאבים וזמן: תכנן עבודה מדויק עם ניהול משימות (Trello), חלוקת עבודה יעילה בצוות, ומיקוד בפיתוח הפיצ'רים החשובים ביותר (MVP) תחילה.

תוכניות "מגירה" למקרה ואחד הסיכונים קורה:

- במקרה של בעיות בגישה לנתונים האמיתיים, נמשיך בפיתוח עם נתוני דמה מפורטים וניצור אבטיפוס להדגמה שתתמוך בהרחבת הפרויקט בעתיד.
- אם יופיעו עיכובים, נדחה פיצ'רים משניים לשלב עתידי, ונתמקד בהשלמת הליבה התפקודית של האפליקציה במסגרת הזמן.

חלק 3 - דו"ח מחקר היתכנות

היתכנות טכנית

הפרויקט נחשב בר-ביצוע מבחינה טכנית בהתבסס על הטכנולוגיות והמשאבים הזמינים במסגרת הקורס. הפיתוח יתבצע באמצעות Android Studio, פלטפורמה יציבה ונפוצה לפיתוח אפליקציות מובייל, המתאימה לדרישות ומספקת כלים מובנים לניהול הרשאות, לשימוש בחיישני הטלפון ולשילוב התראות.

בנוסף, שימוש ב-Firebase לצורך Authentication, ניהול מסד נתונים (במידת הצורך, מאחר שנרצה להשתמש במסד של האוניברסיטה) ושליחת Push Notifications מאפשר התממשקות קלה, מהירה ואמינה — ללא צורך בשרת עצמאי או ידע מתקדם בתשתיות. הכלים הללו זמינים בחינם ומתאימים למסגרת הקורס.

מבחינת מנוע הגרציה של מערכות השעות, ניתן ליישם אלגוריתמים פשוטים יחסית (כמו backtracking או heuristic search) באופן שתואם את דרישות הפרויקט. אף שהרחבה עתידית עשויה לדרוש משאבים נוספים, הגרסה הבסיסית ניתנת לפיתוח במסגרת הזמן והיכולות של צוות הסטודנטים.

אומנם נבצר ממנו להשתמש בתוכנות עם תשלום, אך קיימים היום הרבה אמצעים למימוש הפרויקט בהצלחה באמצעות הכלים הקיימים, תוך התאמה לתשתיות זמינות, מגבלות פיתוח, ולצרכי הפרויקט כפי שהוגדרו.

היתכנות תפעולית

הפרויקט מתאים באופן טבעי לתהליכים הארגוניים הקיימים באוניברסיטה, ויכול להשתלב בצורה חלקה במערכת רישום הקורסים הנוכחית. האפליקציה מציעה כלים שיכולים לשפר את היעילות התפעולית הן עבור הסטודנטים והן עבור צוותי המידע האקדמי.

מצד הסטודנטים, האפליקציה מפשטת בצורה משמעותית את תהליך בניית מערכת השעות — תהליך שבדרך כלל דורש זמן רב, ניסיונות רבים והתמודדות עם התנגשויות בקורסים. בכך היא מפחיתה עומס על שירותי התמיכה האקדמיים ומקטינה את מספר הפניות למזכירות המחלקות בתקופות רישום עמוסות.

מצד האוניברסיטה, שילוב האפליקציה כ-plugin למערכות קיימות מאפשר שימוש יעיל יותר בנתוני הקורסים והמשאבים. באמצעות סטטיסטיקות מובנות, האפליקציה יכולה לסייע בזיהוי עומסים על קורסים מסוימים, הבנת ביקוש בזמן אמת, ושיפור תכנון היצע הקורסים העתידי. בנוסף, תהליכי ההתראה על מקומות שהתפנו מקטינים עומס על המערכת ומייעלים את תהליך הרישום המאוחר.

לכן, האפליקציה מציעה שיפור ממשי בתהליכים האופרטיביים של האוניברסיטה, ללא צורך בשינויים מהותיים בתשתיות — מה שהופך את הפרויקט לבעל היתכנות תפעולית גבוהה.

היתכנות פיננסית

הנחות בסיס:

צוות הפיתוח כולל 3 אנשים, כל אחד עובד כ-9 שעות בשבוע במשך 12 שבועות, סה"כ כ-324 שעות.

עלות שעה: \$100.

עלות פיתוח כוללת: $324 \times 100 = \$32,400$.

Firebase - עלות תחזוקה שנתית: \$120.

עלות שרת חיצוני לאחסון: \$50 בשנה (הערכה).

עלות כוללת שנתית לתחזוקה: $120 + 50 = \$170$.

מחיר מכירה לאוניברסיטה לשנה: \$5,000.

שיעור הדיסקונט 10% (לפי בדיקה שעשינו שיעור הדיסקונט של 10% הוא ערך נפוץ ומשוערך במודלים פיננסיים עבור פרויקטים טכנולוגיים קטנים ובינוניים, במיוחד באקדמיה או פרויקטים עם סיכון מתון).

מכירות:

בהנחות שיווק וחדירה לשוק, הצפי הוא למכור לאוניברסיטה אחת בשנה הראשונה, לשתיים בשנה השנייה ולארבע בשנה השלישית. כך שהרווח הנקי השנתי צפוי להיות:

- שנה 1: 1 אוניברסיטה - הכנסה \$5,000, תחזוקה \$170
- שנה 2: 2 אוניברסיטאות - הכנסה \$10,000, תחזוקה \$340
- שנה 3: 4 אוניברסיטאות - הכנסה \$20,000, תחזוקה \$680

כעת נחשב את הערך הנוכחי (Present Value) של כל תזרים לפי שיעור הדיסקונט של 10%.

- שנה 1: 1 אוניברסיטה — הכנסה נטו \$4,830
- שנה 2: 2 אוניברסיטאות — הכנסה נטו \$9,660
- שנה 3: 4 אוניברסיטאות — הכנסה נטו \$19,320
- שנה 4: 4 אוניברסיטאות — הכנסה נטו \$19,320 (נניח שמספר האוניברסיטאות נשאר קבוע)

חישוב הערך הנוכחי (PV) לכל שנה בשיעור הדיסקונט 10%:

- שנה 0 (השקעה): $-32,400\$$
- שנה 1: $4,830 / 1.10 = \$4,391$
- שנה 2: $9,660 / 1.10^2 \approx \$7,975$
- שנה 3: $19,320 / 1.10^3 \approx \$14,489$
- שנה 4: $19,320 / 1.10^4 \approx \$13,172$

סכום הערכים הנוכחיים:

$$NPV = -32,400 + 4,391 + 7,975 + 14,489 + 13,172 = 7,627\$$$

חישוב ROI

- הוצאה כוללת: השקעת פיתוח ראשונית של $32,400\$$ + תחזוקה שנתית (אנחנו נחשב אותה בנפרד)
- רווח נקי: נניח רווח מצטבר לאורך 4 שנים בהתבסס על תזרימי המזומנים נטו (הכנסות פחות תחזוקה), ללא הון (גם בשיעור בחישוב זה לא התחשבנו בהון).

סה"כ רווח מצטבר ב-4 שנים:

- שנה 1: $4,830\$$
- שנה 2: $9,660\$$
- שנה 3: $19,320\$$
- שנה 4: $19,320\$$

$$4,830 + 9,660 + 19,320 + 19,320 = 53,130\$ = \text{סה"כ רווח מצטבר}$$

$$ROI = (53,130 - 32,400) / 32,400 \times 100 = (20,730 / 32,400) \times 100 \approx 63.9\%$$

פרק 4 – ניתוח מחזיקי עניין

רשימת בעלי עניין ותחומי עניין וציפיות

1. המרצה והמתרגלים בקורס הנדסת תוכנה באוניברסיטת אריאל

תחומי עניין וציפיות:

הם מצפים שהפרויקט יעמוד בדרישות הקורס, יהיה מתועד היטב, כולל אפיון מלא, תכנון, פיתוח, בדיקות והצגה מקצועית. הם מעוניינים בפרויקט איכותי המציג חשיבה הנדסית, פתרון בעיות, תכנון נכון של ארכיטקטורה ואלגוריתמים, ועמידה בלוחות זמנים.

2. צוות הפיתוח (צוות הסטודנטים, אנחנו)

תחומי עניין וציפיות:

חברי הצוות מצפים לשיתוף פעולה יעיל, חלוקת משימות הוגנת וברורה, תמיכה מקצועית מהמרצים, ותהליך פיתוח מסודר. הציפייה המרכזית היא להצליח להפיק אפליקציה איכותית שעומדת בקריטריוני הקורס ושיש בה חידוש, תוך רכישת ניסיון אמיתי בפיתוח תוכנה, אפיון מערכות ויישום אלגוריתמים.

3. משתמשי קצה – סטודנטים באוניברסיטאות בישראל

תחומי עניין וציפיות:

הסטודנטים רוצים אפליקציה לסידור מערכת שעות נוחה לשימוש ופשוטה להבנה, שתסדר מערכת טובה ויעילה. הם מצפים שהאפליקציה תחסוך זמן, תציע חלופות אוטומטיות, תהיה מהירה ואינטואיטיבית, ותעזור להם לבחור את המערכת המושלמת על בסיס אילוצים אישיים.

4. מוסדות הלימוד (אוניברסיטאות ומכללות)

תחומי עניין וציפיות:

המוסדות מעוניינים במוצר אמין, יציב וניתן להתאמה למערכות קיימות. הם מצפים לקבל כלי שמסייע לשפר את חוויית הרישום לקורסים, להפחית עומסים ולשפר ניהול נתוני ביקוש. חשוב להם שהמערכת תהיה מאובטחת, נתמכת לאורך זמן וניתנת להרחבה בעתיד.

5. משתמשי ממשק הניהול (רכזים אקדמיים, מנהלי מחלקות)

תחומי עניין וציפיות:

בעלי עניין אלה מצפים לקבל גישה פשוטה וברורה לנתונים — כמו מספרי נרשמים, עומסים, שעות מבוקשות וסטטיסטיקות נוספות. הם מעוניינים שהממשק יציג מידע עדכני ואמין שיעזור להם לקבל החלטות אקדמיות ותפעוליות.

6. מתרגלי הקורס או מומחים באלגוריתמיקה(ייעוץ אלגוריתמי/ תכנותי)

תחומי עניין וציפיות:

העניין שלהם בפרויקט הוא לעזור לנו ליצור פרויקט איכותי ויעיל, באמצעות מתן ליווי מקצועי בבחירת שיטות אופטימיזציה מתאימות, הכוונה בנוגע למימוש יעיל של האלגוריתמים, עזרה בזיהוי כשלים אפשריים והבטחת איכות הפתרון האלגוריתמי. הם מצפים מאיתנו בהתאם להדרכה שנתנו לנו: להציג יישום תקין, להראות בחירות אלגוריתמיות נכונות, להבנה באלגוריתמים ושיתוף פעולה.

תוכנית תקשורת

- המרצה והמתרגלים
תקשורת באמצעות הצגות בכיתה, העלאת מטלות סדירה בהתאם לנדרש, פניה בכיתה או במייל לעזרה וייעוץ במקרה הצורך, הצגת אמצע סמסטר והגשת הפרויקט בסוף הסמסטר.
- צוות הפיתוח
תקשורת יומיומית דרך הווטסאפ בנוסף לפגישות שבועיות בזום בהתאם לדרישת המשימות. בנוסף תיאום לניהול משימות דרך trello ועבודה משותפת דרך גיטהאב.
- משתמשי הקצה
התקשורת איתם תתבצע בעיקר בשלבי בדיקות המשתמש. יבוצעו סקרים קצרים ומשוב לאחר שימוש בגרסה ראשונית של האפליקציה.
- מוסדות לימוד
התקשורת איתם תתבצע בשלב התחלתי בעת הבקשה לשימוש במסד הנתונים, ובהמשך במקרה של בעיות. התקשורת עם אוניברסיטאות אחרות תתבצע בהמשך במקרה ונרצה להרחיב את האפליקציה לאוניברסיטאות אחרות.
- משתמשי ממשק הניהול
הם יקבלו עדכונים לקראת סיום הפיתוח, בשלב בו יוצג ממשק הניהול ויתקיים איסוף פידבק על הצגת הנתונים ועל הזרימה במערכת.

- מומחי אלגוריתמים / מתרגלים

התקשורת תהיה לפי צורך, לרוב במספר נקודות קריטיות: בחירת האלגוריתם, בדיקת איכות, תכנון מבני נתונים וסקירת קוד. יתבצעו פגישות קצרות בזום/ פרונטלית בכיתה.