

מערכת מסחר אלקטרוני לרהיטים: התפתחות העיצוב והטמעה

תפיסת העיצוב הראשונית

החזון המקורי של הפרויקט היה ליצור פלטפורמת מסחר אלקטרוני מקיפה למכירת רהיטים, תוך התמקדות ב:

1. אדריכלות backend עוצמתית
2. אימות משתמש מאובטח
3. ניהול מוצרים גמיש
4. עיבוד יעיל של עגלת קניות והזמנות

טרנספורמציות בעיצוב במהלך היישום

✓ דפוס האדריכלות

תפיסה ראשונית: חשבנו בהתחלה להשתמש רק בדפוס Factory ובארכיטקטורת MVC בסיסית
עיצוב מיושם: MVC: משופר עם שכבת Service ודפוס Repository, תוך שימוש במספר דפוסי עיצוב מתקדמים

- הוספת שכבות שירות להפרדת לוגיקה עסקית מגישה לנתונים
- יישום הפרדת תחומים מדויקת יותר
- הוספת דפוס יצרן (Factory) ליצירת אובייקטים דינמיים

✓ אימות משתמש ואבטחה

תפיסה ראשונית: אימות בסיסי באמצעות דוא"ל/סיסמה

עיצוב מיושם: מודל אבטחה משופר

- יישום גיבוב סיסמאות מאובטח עם מלח (salt)
- הוספת אסימוני אימות מבוססי JWT
- יצירת תפקידי משתמש עמידים (לקוח, מנהל, מנהל)
- פיתוח מנגנוני אימות והרשאה מקיפים

✓ ניהול מוצרים

תפיסה ראשונית: פעולות CRUD פשוטות למוצרים **עיצוב מיושם:** מידול מוצרים מתקדם

- יצירת מודל מוצר מבוסס הורשה עם סוגי רהיטים מיוחדים
- יישום אסטרטגיות הנחה דינמיות
- הוספת מאפייני מוצר מקיפים לקטגוריות רהיטים שונות
- פיתוח דפוס יצרן ליצירת מוצרים

✓ עיבוד עגלה והזמנות

תפיסה ראשונית: פונקציונליות בסיסית של עגלה והזמנות

עיצוב מיושם: ניהול הנחות והזמנות מתקדם

- יישום אסטרטגיות הנחה מרובות :
 - הנחה באחוזים
 - הנחת "קנה אחד - קבל אחד חינם"
 - הנחה לרכישה בכמות גדולה
- הוספת דפוס תצפיתן (Observer) להתראות הזמנות
- יצירת מעקב והזמנות מקיף וניהול סטטוס

✓ אינטראקציה עם מסד נתונים

תפיסה ראשונית: שאילתות מסד נתונים ישירות

עיצוב מיושם: גישה למסד נתונים מופשטת ומאובטחת

- יצירת מודול מרכזי לחיבור וביצוע שאילתות מסד נתונים
- יישום טיפול בשגיאות ורישום
- שימוש בשאילתות מפורמטות למניעת SQL injection
- הוספת תמיכה במשיכת ועיבוד נתונים מורכבים

אתגרי יישום מרכזיים ופתרונות

אבטחת אימות

- יישום גיבוב סיסמאות עם מלח
- יצירת אסימוני אימות ואימות מאובטחים
- הוספת בקרת גישה מבוססת תפקידים

שיקולי ביצועים

- שימוש בשאילתות SQL יעילות
- יישום ניהול חיבורי מסד נתונים
- יצירת שיטות משיכת נתונים מיטביות

סקלביליות

- עיצוב ארכיטקטורת שירות מודולרית
- יצירת מחלקות בסיס מופשטות להרחבה
- יישום דפוס יצרן ליצירת אובייקטים גמישה

דפוס עיצוב - מהרעיון למימוש

בתחילת הפרויקט, חשבנו להשתמש בעיקר בדפוס Factory, אך במהלך היישום גילינו צורך במספר דפוס עיצוב מתקדמים:

✓ (Factory Pattern)

- **מיקום** FurnitureFactory: furniture.py-
- **מטרה**: יצירת אובייקטי רהיטים דינמית עם מאפיינים שונים
- **יישום**: יצירת אובייקטים שונים של רהיטים (כיסא, שולחן, ספה) בצורה גמישה

• (Strategy Pattern)

- **מיקום** cart_service.py: :
- **מטרה**: יישום אסטרטגיות הנחה שונות
- **דוגמאות** :

- PercentageDiscount
- BuyOneGetOneDiscount
- BulkDiscount

• (Observer Pattern)

- **מיקום** checkout_service.py: :
- **מטרה**: ניהול והתראות על סטטוס הזמנות
- **רכיבים** :

- ❖ OrderSubject
- ❖ OrderObserver
- ❖ EmailNotification
- ❖ InventoryUpdate

• (Repository Pattern)

- **מיקום**: מודלים שונים כמו user.py, furniture.py
- **מטרה**: הפשטת גישה למסד הנתונים
- **יישום**: שיטות סטטיות לביצוע פעולות CRUD על ישויות שונות

- **(Abstract Factory Pattern)**

- **מיקום:** FurnitureFactory ב: furniture.py-
- **מטרה:** יצירת אובייקטים מסוגים שונים באופן דינמי
- **יישום:** יצירת אובייקטי רהיטים שונים עם מאפיינים ייחודיים

- **(Builder Pattern)**

- **מיקום:** חלקים שונים בקוד
- **מטרה:** בניית אובייקטים מורכבים בצעדים
- **דוגמה:** בניית הזמנות מורכבות ב OrderService-

הבשלת תפיסת העיצוב

מהרעיון הראשוני של דפוס Factory הבודד, הפרויקט התפתח לשימוש במספר דפוסי עיצוב מתקדמים שסייעו ביצירת קוד גמיש, ניתן להרחבה ומאורגן.

שיפורים עתידיים

- יישום רישום (logging) מקיף יותר
- הוספת מנגנוני קאשינג מתקדמים יותר
- שיפור טיפול בשגיאות ומשוב למשתמש
- פיתוח אסטרטגיות הנחה ותמחור מתוחכמות יותר

סיכום

היישום התפתח מרעיון מסחר אלקטרוני בסיסי לפלטפורמת מכירת רהיטים עמידה וניתנת להרחבה עם תכונות מתקדמות ועקרונות עיצוב מאובטחים.