

אפיון אפליקציית חנות משחקים אינטרנטית

מסמך זה מפרט את הדרישות והמאפיינים של אפליקציית חנות משחקים אינטרנטית, תוך התמקדות בטכנולוגיות שנבחרו: Django לצד השרת ו-PostgreSQL כבסיס נתונים.

1. סקירת פרויקט

אפליקציית חנות המשחקים האינטרנטית תספק פלטפורמה למכירה ורכישה של משחקים דיגיטליים. היא תכלול ממשק ניהול מלא למנהל האתר (המוכר) וממשק משתמש עבור הלקוחות. המטרה היא ליצור מערכת קלה לשימוש, מאובטחת ויעילה לניהול מלאי משחקים, משתמשים והזמנות.

2. פיצ'רים מרכזיים

האפליקציה תכלול שני אזורים עיקריים: פאנל ניהול (Admin Panel) וממשק חנות (Storefront).

2.1. פאנל ניהול (Admin Panel)

פאנל הניהול יאפשר למנהל האתר שליטה מלאה על תוכן החנות, משתמשים והזמנות.

- **ניהול משחקים:**
 - **הוספת משחק חדש:**
 - שם המשחק (טקסט)
 - תיאור מפורט (טקסט ארוך)
 - מחיר (מספר עשרוני, עם אפשרות למטבע)
 - קטגוריה/ז'אנר (בחירה מתוך רשימה קיימת ואפשרות להוספת חדשים)
 - תמונות (אפשרות להעלות מספר תמונות, כולל תמונה ראשית/כיסוי)
 - סרטוני וידאו (הטמעת קישורי YouTube/Vimeo)
 - קובץ משחק/קישור להורדה (עבור משחקים דיגיטליים)
 - תאריך יציאה
 - פלטפורמות נתמכות (לדוגמה: PC, Xbox, PlayStation)
 - דרישות מערכת (מינימום ומומלץ - טקסט חופשי)
 - סטטוס זמינות (לדוגמה: "זמין במלאי", "אזל", "בקרוב")
 - **עריכת משחק קיים:** עדכון כל הפרטים של משחק קיים.
 - **הסרת משחק:** מחיקת משחק מהחנות.
 - **חיפוש וסינון:** אפשרות לחפש ולסנן משחקים לפי שם, קטגוריה, מחיר ועוד.
- **ניהול משתמשים והרשאות:**
 - **רשימת משתמשים:** הצגת כל המשתמשים הרשומים במערכת.
 - **עריכת פרטי משתמש:** עדכון פרטים אישיים, איפוס סיסמה.
 - **הגדרת הרשאות:**
 - **מנהל (Admin):** גישה מלאה לפאנל הניהול.

■ **משתמש רגיל (User):** גישה לאזור האישי, רכישת משחקים.

■ אפשרות עתידית להוספת רמות הרשאה נוספות.

○ **חסימה/מחיקה של משתמשים.**

● **ניהול הזמנות (אופציונלי בשלב ראשון, מומלץ לשלב שני):**

○ צפייה בכל ההזמנות שבוצעו.

○ עדכון סטטוס הזמנה (לדוגמה: "ממתין לתשלום", "שולם", "בוטל").

○ צפייה בפרטי הזמנה מלאים.

2.2. ממשק חנות (Storefront)

ממשק החנות הוא הממשק שבו המשתמשים יתקשרו עם האפליקציה.

● **דף הבית:**

○ הצגת משחקים מומלצים, חדשים, או במבצע.

○ באגרים פרסומיים/קרוסלות.

● **ניווט וחיפוש:**

○ ניווט לפי קטגוריות/ז'אנרים.

○ פונקציונליות חיפוש מתקדמת (לפי שם, מחיר, פלטפורמה, וכו').

● **דף פרטי משחק:**

○ הצגת שם, תיאור, מחיר, תמונות, וסרטונים.

○ הצגת דרישות מערכת.

○ כפתור "הוסף לעגלה".

○ אפשרות לביקורות ודירוגים (אופציונלי).

● **עגלת קניות:**

○ הצגת פריטים שנבחרו.

○ אפשרות להסיר פריטים או לשנות כמות.

○ סיכום ביניים של ההזמנה.

● **תהליך תשלום (Checkout):**

○ בחירת אמצעי תשלום (יש לשלב ספק שירותי סליקה צד שלישי).

○ הזנת פרטי תשלום מאובטחים.

○ אישור וסיום הזמנה.

● **אזור אישי למשתמש:**

○ היסטוריית רכישות.

○ גישה להורדות של משחקים שנרכשו.

○ עדכון פרטים אישיים ושינוי סיסמה.

○ הרשמה והתחברות משתמשים.

3. מחסנית טכנולוגית (Technology Stack)

● **צד שרת (Backend):** Python עם Django Framework.

○ **למה Django:** Django היא פלטפורמת פיתוח ווב חזקה ומלאה (full-stack)

ב-Python. היא כוללת מודל (ORM (Object-Relational Mapper מובנה, מערכת

ניהול משתמשים (Authentication and Authorization), ופאנל ניהול אוטומטי (Django Admin) שיתאים באופן מושלם לניהול המשחקים והמשתמשים שלך. היא מקדמת פיתוח מהיר, אבטחה מובנית וסקלביליות.

- **צד לקוח (Frontend):** HTML, CSS, JavaScript באמצעות מערכת התבניות של Django (Jinja2) או מנוע התבניות המובנה).

- **למה כך:** גישה זו מאפשרת פיתוח מהיר ואינטגרציה חלקה עם ה-Backend של Django, ללא צורך ב-Framework נפרד של JavaScript בשלב ראשון, מה שמפשט את תהליך הפיתוח.

- **בסיס נתונים (Database):** PostgreSQL.

- **למה PostgreSQL:** זהו בסיס נתונים יחסי (SQL) בקוד פתוח, חזק, אמין ובעל יכולות מתקדמות רבות. הוא מתאים במיוחד לאפליקציות הדורשות עקביות נתונים, תמיכה בעסקאות מורכבות, וסקלביליות. יש לו תמיכה מצוינת ב-Django והוא נחשב לסטנדרט תעשייתי עבור יישומים בסדר גודל כזה. הוא חנמי לחלוטין לשימוש.

4. סכמת בסיס נתונים (קונספטואלית)

להלן טבלאות הליבה שיהיו בבסיס הנתונים:

- **משתמשים (Users):**

- id (מזהה ייחודי)
- username (שם משתמש)
- email (כתובת מייל)
- password_hash (סיסמה מוצפנת)
- is_admin (בולאני, האם המשתמש הוא מנהל)
- date_joined (תאריך הצטרפות)

- **משחקים (Games):**

- id (מזהה ייחודי)
- name (שם המשחק)
- description (תיאור)
- price (מחיר)
- category_id (מפתח זר לטבלת קטגוריות)
- release_date (תאריך יציאה)
- download_link (קישור להורדה/קובץ)
- status (סטטוס זמינות)

- **קטגוריות (Categories):**

- id (מזהה ייחודי)
- name (שם הקטגוריה, לדוגמה: "פעולה", "הרפתקה")

- **תמונות משחק (GameImages):**

- id (מזהה ייחודי)
- game_id (מפתח זר לטבלת משחקים)

- image_url (קישור לתמונה)
- is_main (בולאני, האם זו תמונת הכיסוי)
- **סרטוני משחק (GameVideos):**
- id (מזהה ייחודי)
- game_id (מפתח זר לטבלת משחקים)
- video_url (קישור לסרטון, לדוגמה מיוטיוב)
- **פלטפורמות (Platforms):**
- id (מזהה ייחודי)
- name (שם הפלטפורמה, לדוגמה: "PC", "Xbox")
- **Game_Platforms (טבלת קישור רבים-לרבים):**
- game_id (מפתח זר לטבלת משחקים)
- platform_id (מפתח זר לטבלת פלטפורמות)
- **הזמנות (Orders):**
- id (מזהה ייחודי)
- user_id (מפתח זר לטבלת משתמשים)
- order_date (תאריך הזמנה)
- total_amount (סכום כולל)
- status (סטטוס הזמנה)
- **פריטי הזמנה (OrderItems):**
- id (מזהה ייחודי)
- order_id (מפתח זר לטבלת הזמנות)
- game_id (מפתח זר לטבלת משחקים)
- price_at_purchase (מחיר המשחק בעת הרכישה)

5. מדריך התקנה והגדרה

להלן השלבים להתקנה והגדרה של סביבת הפיתוח.

5.1. התקנת Python

1. **הורדת Python:**
 - כנס לאתר הרשמי של Python: <https://www.python.org/downloads>
 - הורד את הגרסה היציבה האחרונה (מומלץ Python 3.9+).
2. **התקנה:**
 - הפעל את קובץ ההתקנה.
 - **חשוב מאוד:** וודא שאתה מסמן את התיבה "Add Python X.Y to PATH" במהלך ההתקנה (ב-Windows). זה יאפשר לך להריץ פקודות Python מכל מקום בטרמינל.
 - בצע את ההוראות שעל המסך להשלמת ההתקנה.
3. **אימות התקנה:**
 - פתח טרמינל (CMD ב-Windows, Terminal ב-macOS/Linux).
 - הקלד: `python --version` (או `python3 --version` ב-macOS/Linux).

- אתה אמור לראות את גרסת הפיתון שהתקנת.

5.2. התקנת PostgreSQL

1. הורדת PostgreSQL:

- כנס לאתר הרשמי של PostgreSQL: <https://www.postgresql.org/download/>
- בחר את מערכת ההפעלה שלך ופעל לפי ההוראות להורדת המתקין.

2. התקנה:

- הפעל את קובץ ההתקנה.
- במהלך ההתקנה, תתבקש להגדיר סיסמה למשתמש הראשי postgres. **שמור את הסיסמה הזו במקום בטוח!**
- תתבקש לבחור רכיבים להתקנה. וודא ש-"PostgreSQL Server" ו-"pgAdmin" (כלי ניהול גרפי) מסומנים.
- השלם את תהליך ההתקנה.

3. יצירת בסיס נתונים חדש:

- פתח את pgAdmin (מותקן עם PostgreSQL).
- התחבר לשרת PostgreSQL באמצעות הסיסמה שהגדרת.
- לחץ קליק ימני על "Database" -> "Create" -> "...Databases".
- תן שם לבסיס הנתונים שלך (לדוגמה: game_store_db).
- וודא שה-Owner הוא postgres (או משתמש אחר שיצרת).
- לחץ "Save".

5.3. הגדרת סביבת פיתוח ו-Django

1. יצירת סביבה וירטואלית (Virtual Environment):

- נווט לתיקייה שבה תרצה ליצור את הפרויקט שלך בטרמינל.
- הקלד: `python -m venv venv` (זה ייצור תיקייה בשם venv שתכיל את הסביבה הווירטואלית).

2. הפעלת הסביבה הווירטואלית:

- **Windows:** `venv\Scripts\activate`
- **macOS/Linux:** `source venv/bin/activate`
- תראה (venv) בתחילת שורת הפקודה, מה שמעיד שהסביבה הווירטואלית פעילה.

3. התקנת Django ו-psycopg2 (מנהל התקן ל-PostgreSQL):

- ודא שהסביבה הווירטואלית פעילה.
- הקלד: `pip install Django psycopg2-binary`

4. יצירת פרויקט Django חדש:

- ודא שהסביבה הווירטואלית פעילה.
- הקלד: `django-admin startproject game_store_project` . (הנקודה בסוף תיצור את הפרויקט בתיקייה הנוכחית).

5. הגדרת בסיס הנתונים ב-Django:

- פתח את הקובץ `game_store_project/settings.py`.

- מצא את החלק DATABASES ושנה אותו כך:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'game_store_db', # שם pgAdmin בסיס הנתונים שיצרת ב
        'USER': 'postgres', # שם המשתמש של PostgreSQL (דרך כלל)
        'PASSWORD': 'your_postgres_password', # הסיסמה שהגדרת להתקנת
    }
    PostgreSQL
    'HOST': 'localhost', # בדרך כלל 127.0.0.1
    'PORT': '5432', # PostgreSQL פורט ברירת המחדל של
}
```

- **חשוב:** החלף את 'your_postgres_password' בסיסמה האמיתית שלך.

6. יצירת אפליקציות Django:

- נווט לתיקיית הפרויקט (שם נמצא manage.py).
- הקלד: `python manage.py startapp games`
- הקלד: `python manage.py startapp users`
- הקלד: `python manage.py startapp admin_panel` (או כל שם אחר שתבחר למודול הניהול)

7. הוספת האפליקציות ל-settings.py:

- בתוך `game_store_project/settings.py`, הוסף את שמות האפליקציות שיצרת ל-`INSTALLED_APPS`:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'games', # האפליקציה שיצרת
    'users', # האפליקציה שיצרת
    'admin_panel', # האפליקציה שיצרת
    # ... אפליקציות נוספות בעתיד
]
```

8. ביצוע מיגרציות (Migrations) לבסיס הנתונים:

- ודא שהסביבה הווירטואלית פעילה.
- הקלד: `python manage.py makemigrations`
- הקלד: `python manage.py migrate` (זה ייצור את הטבלאות בבסיס הנתונים שלך).

9. יצירת משתמש על (Superuser) למערכת הניהול:

- הקלד: `python manage.py createsuperuser`
- פעל לפי ההוראות ליצירת שם משתמש, כתובת מייל וסיסמה למנהל.
- 10. **הפעלת שרת הפיתוח:**
- הקלד: `python manage.py runserver`
- פתח את הדפדפן שלך ונווט לכתובת: `/http://127.0.0.1:8000/admin`
- התחבר עם פרטי המשתמש על שיצרת. זהו פאנל הניהול האוטומטי של Django!

6. שיקולי פריסה (Deployment)

לאחר סיום הפיתוח, האפליקציה תצטרך להיות פרוסה על שרת כדי להיות נגישה לציבור. פלטפורמות מומלצות כוללות:

- **Heroku:** פלטפורמת PaaS (Platform as a Service) קלה לשימוש, עם שכבת חינם להתחלה.
- **PythonAnywhere:** פלטפורמה ייעודית לאירוח יישומי Python, גם היא עם תוכנית חינמית.
- **ספקיות ענן (AWS, Google Cloud, Azure, DigitalOcean):** מציעות שליטה מלאה יותר אך דורשות ידע טכני מתקדם יותר.

7. שיקולי אבטחה

- **אימות והרשאות:** שימוש במערכת ה-Authentication וה-Authorization המובנית של Django.
 - **הצפנת סיסמאות:** Django מטפלת בהצפנת סיסמאות באופן אוטומטי.
 - **הגנה מפני התקפות נפוצות:** Django מספקת הגנות מובנות מפני SQL Injection, XSS, CSRF.
 - **HTTPS:** חובה להשתמש ב-HTTPS (SSL/TLS) בסביבת Production כדי להצפין את התקשורת בין הלקוח לשרת.
 - **תיקוף קלט (Input Validation):** וודא שכל קלט מהמשתמשים עובר תיקוף קפדני לפני עיבודו או שמירתו בבסיס הנתונים.
- מסמך זה מספק בסיס מוצק לתחילת פיתוח אפליקציית חנות המשחקים שלך.