# יש לכלול בתחילת מסמך הפרויקט את דף השער הבא: **טכנולוגיות אינטרנט מתקדמות - 61776 (WEB)**

**הגשת פרויקט**

**<Online Art gallery and marketplace> <B10><קבוצה 6>**

|  |  |
| --- | --- |
| **שם חבר.ת הצוות** | **תז** |
| אביב רז | 207940875 |
| ארתור צ'רני | 206466500 |
| דניאל פלדמן | 208134700 |
| שגיא יוסופוב | 315030155 |

תקציר פרויקט:

הפרויקט שלנו הוא פלטפורמה לאמנים להציג ולמכור את עבודות האומנות שלהם, עם פיצ'ר של תשלום מאובטח.

המשתמשים במערכת שלנו הם מבקר (משתמש לא רשום) אמן, ומנהל.

משתמש לא רשום: יכול להתחבר, להירשם לפלטפורמה בתור אומן או מנהל. בנוסף ניתן לעיין בגלריה ולצפות בפרטים של יצירות אומנות בהרחבה, כולל פרטי אומן. המשתמש יכול להוסיף יצירות אומנות לעגלה ולבצע תשלום מאובטח ע"י PayPal או באמצעות כרטיס אשראי.

אמן: מסוגל לצפות ולקנות כמו כל משתמש לא רשום, ובנוסף בעל דף אישי בו יכול להעלות יצירות חדשות, לקבוע להם מחיר, לשים פירוט ומידות. בנוסף יכול למחוק יצירות שלו ולעדכן את הפרטים של יצירות קיימות.

מנהל: מסוגל לבצע כל מה שמשתמש לא רשום יכול לעשות. ובנוסף בעל דף מנהל בו יכול לצפות בכל המשתמשים ולאפשר גישה או להסיר להם גישה כרצונו.

משתמשים לבדיקת המערכת:

אמן:

שם משתמש 1: daniel\_feldstone שם משתמש 2: artist  
סיסמא 1: 123 סיסמא 2: artist

מנהל:

שם משתמש 1: sagi שם משתמש 2: admin

סיסמא 1: sagi סיסמא 2: admin

**הטכנולוגיות המרכזיות:**

* **Frontend:** Next.js
* **Styling:** Tailwind CSS
* **Backend:** Next.js (API routes)
* **Database:** MongoDB
* **Others:** Vercel (deployment), GitHub (version control), JavaScript (main language)

[קישור לתיקיית גיט ציבורי](https://github.com/SagiYosofov/ArtMarketNextJs)

[קישור לאתר](https://artmarket-final.vercel.app/)

מהנדס המערכת - ארתור צ'רני

חלוקת העבודה נקבעה בתחילת הקורס, כאשר כל חבר צוות קיבל את העבודה שמעניינת אותו ביותר. הממשק ביננו התבצע משך כל הסמסטר, עדכונים באמצעות ווצאפ, ושיתוף פעולה בעזרת הגיטהאב, שאפשר לנו לראות שינויים בקלות, ולמזג עבודות שבוצעו במקביל.

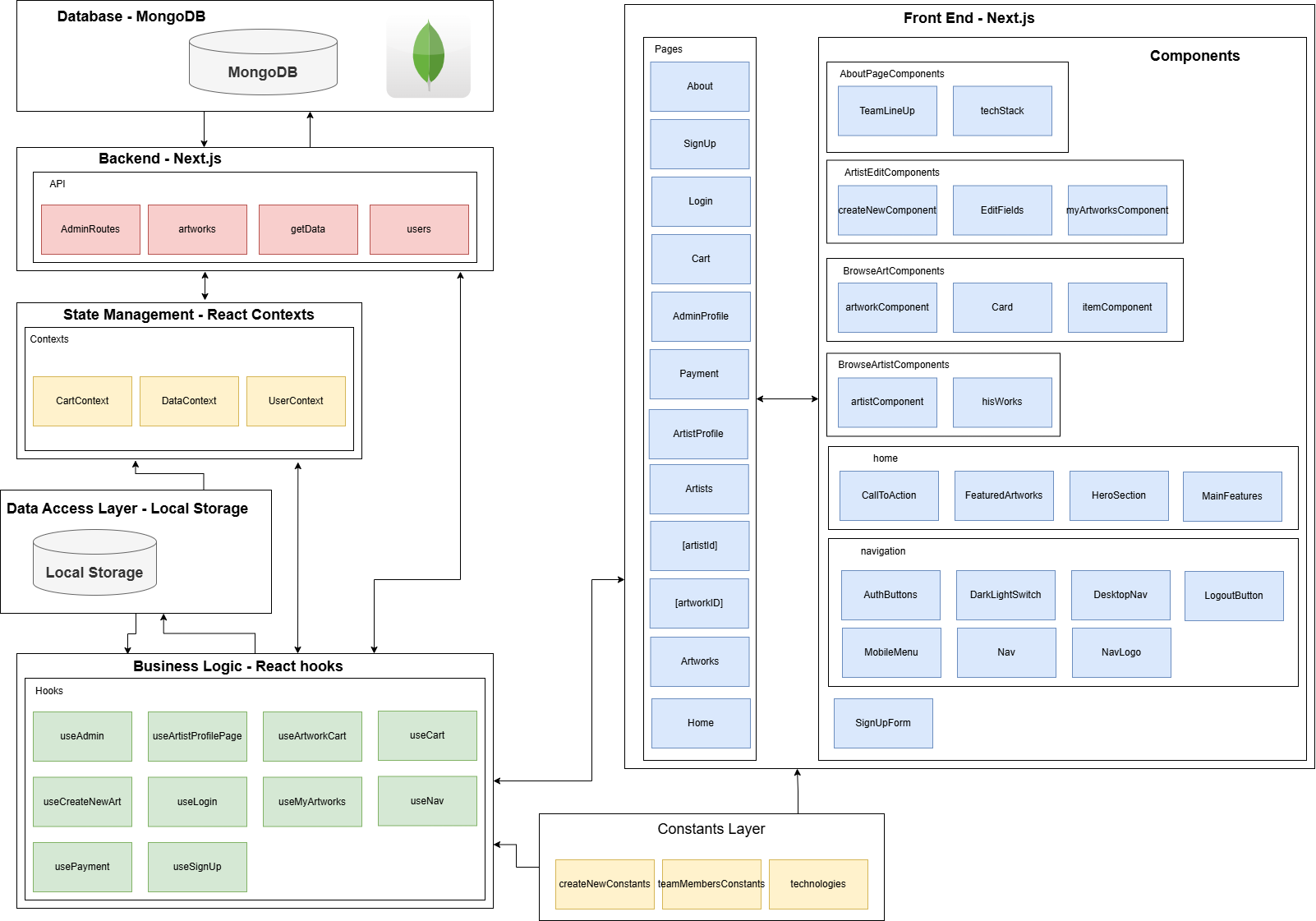
|  |  |  |
| --- | --- | --- |
| **שם חבר הצוות** | **משימות שהוקצו** | **משימות שהושלמו** |
| ארתור צ'רני - מהנדס מערכת | רספונסיביות, מסך about, תשלום מאובטח, מסך אמנים ומסך אמן נבחר, וידוא הערות בקוד | רספונסיביות, מסך about, תשלום מאובטח, מסך אמנים ומסך אמן נבחר, וידוא הערות בקוד, פריסת האתר בורסל |
| דניאל פלדמן | סל קניות, מסך אומנות, מסך אומנות נבחרת, דיאגרמת ארכיטקטורה, דיאגרמת קבצים, וידוא חלוקת פונקציונליות ותכן | סל קניות, מסך אומנות, מסך אומנות נבחרת, דיאגרמת ארכיטקטורה, דיאגרמת קבצים, וידוא חלוקת פונקציונליות ותכן |
| אביב רז | דף בית, דף מנהל וניהול סטטוס משתמשים, בדיקות שימושיות האתר, Dark/Light switch.  פריסת האתר בורסל | דף בית, דף מנהל וניהול סטטוס משתמשים, בדיקות שימושיות האתר, Dark/Light switch. |
| שגיא יוסופוב | הקמת מסד נתונים ותקשורת עם המסד, יצירת תיקייה בגיטהאב, מסך התחברות והרשמה, מודלים של שמירה במסד (משתמש, אמן אומנות) | הקמת מסד נתונים ותקשורת עם המסד, יצירת תיקייה בגיטהאב, מסך התחברות והרשמה, מודלים של שמירה במסד (משתמש, אמן אומנות) |

2. הציגו רשימת דרישות פונקציונליות ולא פונקציונליות (בנפרד, יש לסווג דרישות לא פונקציונליות לפי wikipedia NFR).

**דרישות**

1. **דרישות פונקציונליות FR**
   1. המערכת תאפשר הרשמה
   2. המערכת תאפשר התחברות
   3. המערכת תאפשר צפייה באומנות
   4. המערכת תאפשר צפייה באמנים
   5. המערכת תאפשר הוספה/מחיקה לעגלה
   6. המערכת תאפשר תשלום מאובטח
   7. המערכת תאפשר העלאת אומנות למכירה
   8. המערכת תאפשר מחיקת אומנות
   9. המערכת תאפשר עריכת מידע של אומנות
   10. המערכת תאפשר אישור משתמשים חדשים
   11. המערכת תאפשר נעילת גישה לאתר למשתמשים
   12. המערכת תאפשר לעבור ממצב בהיר/כהה
2. **NFR**
   1. Performance
      1. המערכת תשלים התחברות והרשמה תוך 2 שניות מקסימום.
      2. המערכת תעדכן מסד נתונים עם כל שינוי של משתמש [הוספה,קניה וכו'] מיד עם לחיצת הכפתור.
   2. Accessibility
      1. כפתור מעבר למצב כהה/בהיר יוצג בסרגל עליון שנגיש מכל דף באתר.
   3. Security
      1. המערכת תאפשר העלאת תמונות של יצירות אומנות מספקי אחסון עם מוניטין גבוה בלבד.
      2. במערכת יהיו שני סוגים של משתמשים רשומים: אמן ומנהל.
      3. רק מנהל יוכל לאשר או לנעול משתמשים אחרים.
      4. מנהל לא יוכל לנעול לעצמו את הגישה לאתר.
      5. רק אמן יוכל להעלות,לערוך ולמחוק את פיסות האומנות שבדף שלו.
      6. התשלום המאובטח יהיה דרך PayPal.
   4. Usability
      1. המערכת תציג פידבק לכפתורים חשובים.
      2. אמנים ואומנות יוצגו בכרטיסים (Cards).
      3. למערכת יהיה סרגל עליון לניווט באתר.
      4. השדות בהרשמה יהיו שם משתמש, סיסמא, שם פרטי, שם משפחה, מייל, כתובת וסוג משתמש.
      5. רק אמן יוכל למלא bio בעת רישום.
   5. Adaptability
      1. תצוגת המערכת תהיה אופטימלית לכל מכשיר ללא חשיבות לרזולוציית מסך.
      2. המערכת תציג תפריט סנדוויץ' עבור מסך קטן.
   6. Maintainability
      1. קבצי המערכת יהיו קטנים וממוקדים, לצורך תחזוקה מהירה.
      2. פונקציונליות, ממשק משתמש, קבועים, וapi יופרדו לקבצים ותיקיות שונות.
      3. קומפוננטות יחולקו לתיקיות לפי דף.
   7. Data Integrity
      1. המערכת לא תאפשר רישום של שני משתמשים עם אותו שם משתמש.

3. מבנה ואינטראקציה:  
 א. הציגו ארכיטקטורה מעודכנת של האתר (תרשים הכולל את האלמנטים המרכזיים)



קישור לדיאגרמת ארכיטקטורה במידת הצורך:

[ArtMarket Architecture](https://drive.google.com/file/d/1ssQ8OUV_iOUA2F8b4EchD1gHie4ragSL/view?usp=sharing)

ב.הציגו דיאגרמת use case המתארת את השימוש באתר.



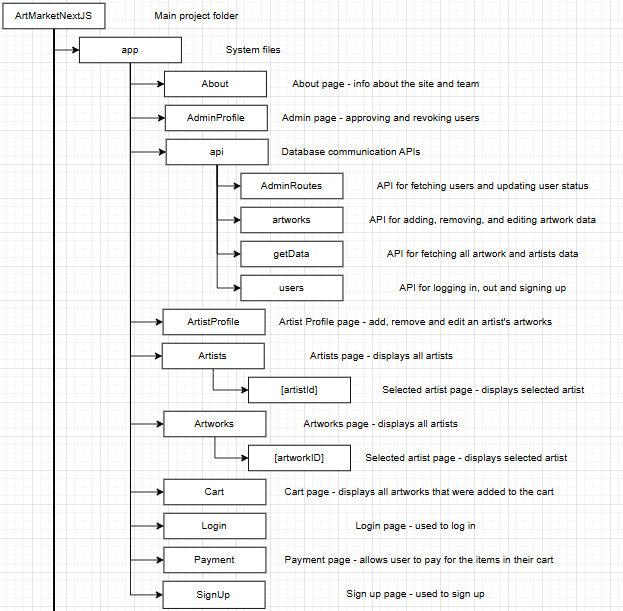
4. יש להציג מבנה סופי של האתר שלכם:

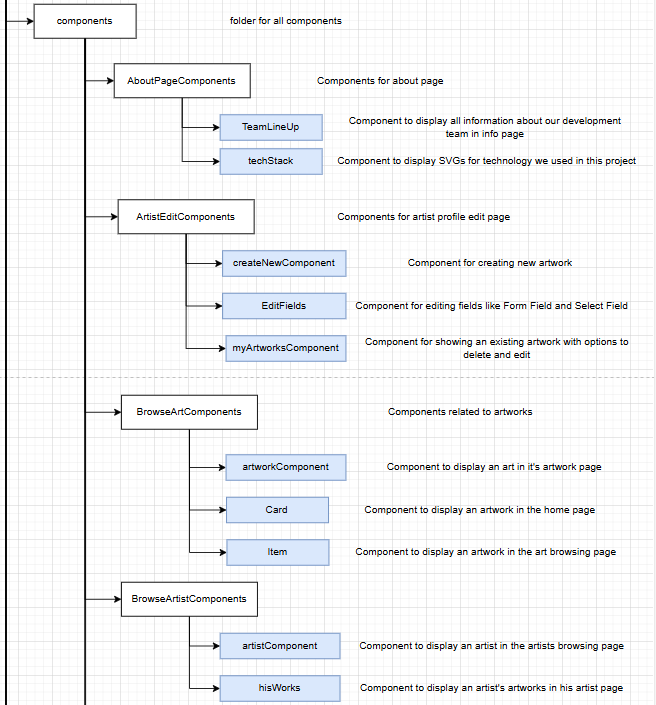
**נדרשת בכל פרויקט פריסה מלאה (deployment) של הפרויקט!**   
**לא יתקבלו הגשות של קבצים או אתר ב -localhost .**

**טכנולוגיות**: להלן המרכיבים הטכנולוגיים המומלצים לשימוש בפרויקט:

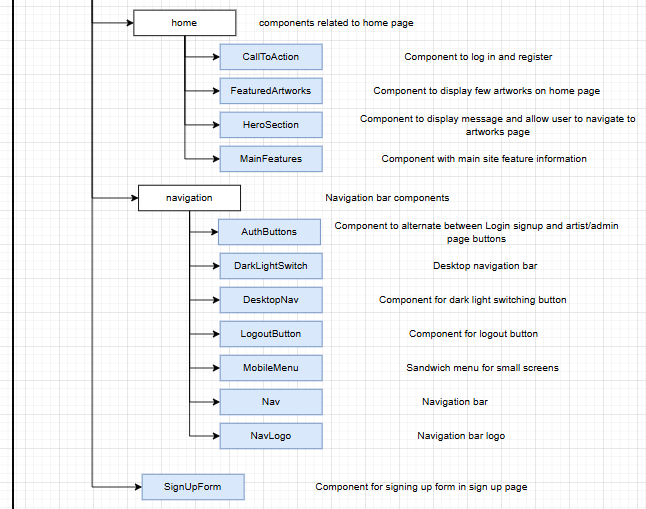
א. האתר ימומש ב -react/preact/Next, וכן שימוש ב Tailwind - נא להציג דיאגרמה המתארת את התיקיות והקבצים השונים. יש לפרט את הקומפוננטות השונות.

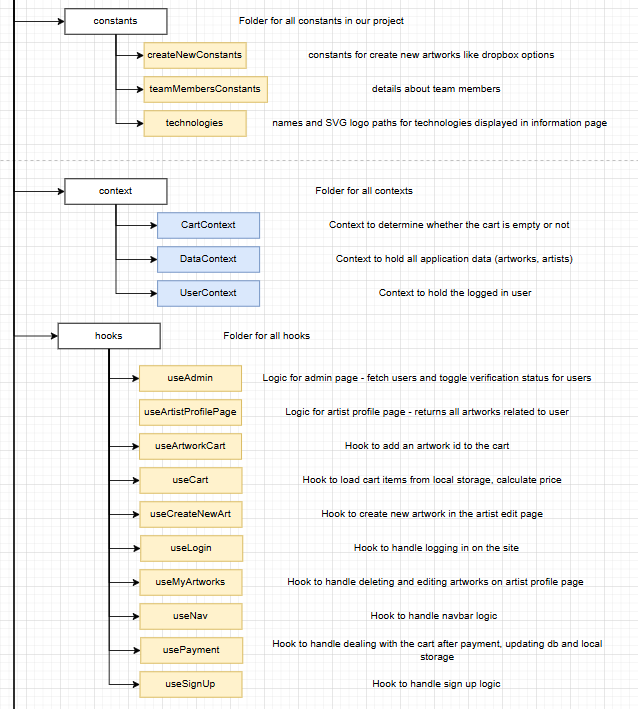
\* נא לשים לב שהדיאגרמה מחולקת למספר דפים עקב גודלה

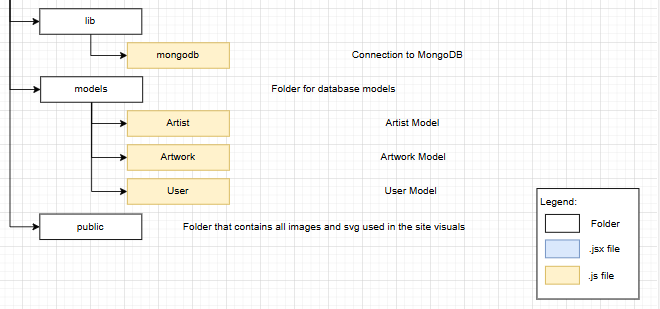




Components related to artists



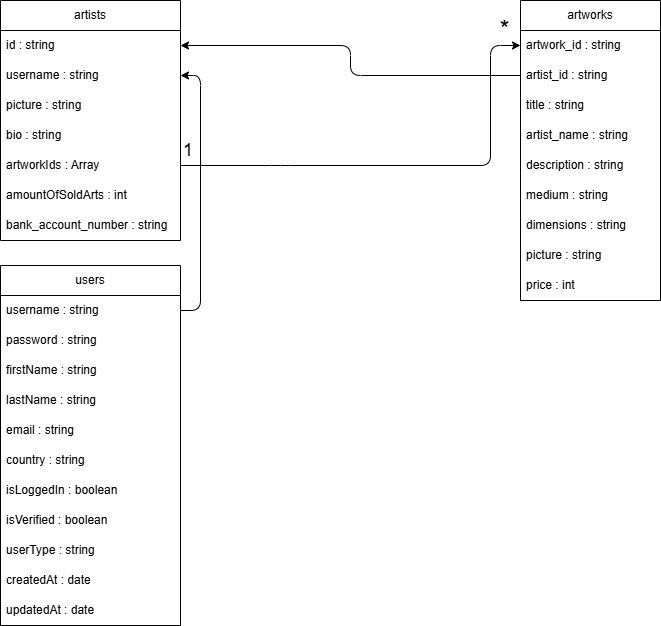




קישור לדיאגרמת קבצים ותיקיות במידת הצורך:

[ArtMarket – Files and Folders Diagram](https://drive.google.com/file/d/1XyAWCV_O4R0_LIVWi80oEqFR-R7VngoT/view?usp=sharing)

ב. יש לפרט את פריטי המידע - יש להשתמש במידע אמיתי ורלוונטי לפרויקט שלכם (בשליפה ממסד נתונים חיצוני או מ - API). יש להראות דיאגרמת מבנה DB.



5. יש להגיש תיק למתכנת:

א. יש לתעד ולהסביר את כל הפונקציות המרכזיות בקוד שלכם.

**Contexts:**

DataContext:

|  |  |
| --- | --- |
| State Variables | Explanation |
| dbUpdate | Used to track if the data in the context needs to be refreshed |
| artistsData | An array that holds all the information about the artists |
| artworksData | An array that holds all the information about the artworks |
| isLoading | A flag that tells if data is being loaded |
| error | Stores error messages |
| Function | Explanation |
| fetchData | Fetches artwork and artist data from mongoDB using /api/getData endpoint and stores them in artistsData and artworksData |
| useEffect | Activates fetchData everytime the dbUpdate state variable is updated. |
| refreshData | Manually triggers a data refresh by setting dbUpdate to true |
| useData | Custom hook to easily access DataContext values |

CartContext:

|  |  |
| --- | --- |
| State Variables | Explanation |
| hasCartItems | A flag that indicates if the cart is empty or not |
| Functions | Explanation |
| useEffect | Updates the cart status every time the cart changes |
| checkCart | Checks if there are items in the cart and updates the hasCartItems flag accordingly |
| useCart | Custom hook to access the CartContext vars and funcs |

UserContext:

|  |  |
| --- | --- |
| State Variables | Explanation |
| user | Stores the logged in user |
| Functions | Explanation |
| useEffect | Retreives user from local storage if it exists on page render |
| useEffect | Update the user in the local storage whenever it changes |
| useUser | Custom hook to use the user context |

**Hooks**:

useAdmin:

|  |  |
| --- | --- |
| Function | Explanation |
| fetchUsers | Sends a request to the server using api and returns a json of all users details except the current user. |
| toggleVerification | Changes the given user’s verification status both on the database and locally. |

useArtistProfilePage:

|  |  |
| --- | --- |
| Function | Explanation |
| checkAuth | Updates the locally saved artworks in case the logged in user is an artist. |

useArtworkCart:

|  |  |
| --- | --- |
| State Variables | Explanation |
| addedToCart | Used as a flag to tell if an item was added to the cart recently |
| Functions | Explanation |
| addToCart | Adds a given artwork data to the cart and raises the added to cart flag for 2 seconds. |

useCart:

|  |  |
| --- | --- |
| State Variables | Explanation |
| cartItems | An array to store all items in the cart |
| totalSum | Stores the sum of the cost of all items in the cart |
| artworksData | Stores all the artworks data, comes from the data context |
| isLoading | Flag that tells if data is being loaded from the database, taken from the data context |
| error | Error message, comes from the data context |
| Functions | Explanation |
| useEffect | Updates the cart items and total price every time the artworks data is updated. |
| handleRemoveItem | Removes an item from the cart and recalculates the total price |
| handleCheckout | Moves the user to the checkout page with the total sum of the cart. |

UseCreateNewArt:

|  |  |
| --- | --- |
| Function | Explanation |
| validateImageUrl | Checks to see if an entered image url during new art creation is valid, meaning it belongs to our allowed hostnames and is an image |
| handleChange | Updates the locally saved form data from the new art forms and displays a message according to input. |
| handleSubmit | Validates input and sends a request to the server, creating a new art in the database with the given art data. |

useLogin:

|  |  |
| --- | --- |
| State Variables | Explanation |
| SERVER\_URL | Database url for mongoDB requests. |
| formData | Stores inputs for the username and password |
| responseMessage | Stores the server’s response message |
| Functions | Explanation |
| handleSubmit | Handles form submission to send login details to server |

useMyArtworks:

|  |  |
| --- | --- |
| Functions | Explanation |
| handleEditClick | Creates a copy of the selected artwork locally when the edit button is clicked |
| handleInputChange | Updates the locally saved artwork with each change that is done to the fields. |
| handleDeleteClick | Removes the selected artwork both from the presentation layer and from the database by using an api call |
| handleSubmit | Updates the displayed artwork and the database with the information given by the user. |

useNav:

|  |  |
| --- | --- |
| Functions | Explanation |
| handleLogout | Sets the user’s logged in status to 0 by using an api call, and removes the current user from the local storage, also deleting the cart |

usePayment:

|  |  |
| --- | --- |
| Functions | Explanation |
| deletePurchasedArtworks | Deletes purchased artworks from the database after a succesfull transaction |
| handleSuccessfulPurchase | Deletes purchased artworks from the database, local storage and notifies the system. |
| handleDemoPayment | Deletes the artworks in the cart from the database and sends the user a message. |

useSignUp:

|  |  |
| --- | --- |
| Functions | Explanation |
| handleChange | Updates the local data every time a field changes |
| handleSubmit | Creates a new user in the database using an api call and resets the fields. |

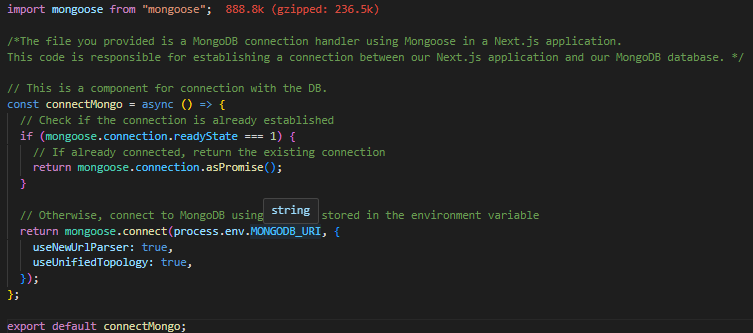
ב. יש לפרט את כל ה -API שהשתמשתם בהם, קישור ל- DB,סביבות מיוחדות שהתקנתם, קטעי קוד מיוחדים שלקחתם ממקורות שונים (כולל רפרנסים) , פרומפטים שהשתמשתם בהם בכלי AI.

**Api:**

|  |  |
| --- | --- |
| AdminRoutes\userManager | |
| Functions |  |
| GET | Connects to mongoDB and fetches a list of all users, without their passwords. |
| PUT | Connects to mongoDB and sends a request to approve a user by changing their status |
| artworks | |
| POST | Adds a given artwork to the database |
| DELETE | Deletes a given artwork from the database |
| deletePurchased |  |
| DELETE | Deletes multiple artworks from the database |
| edit\[id] |  |
| PUT | Updates a given artwork’s data in the database |
| getData | |
| GET | Fetches all needed site data from the database (artworks, artists and users) |
| users | |
| Login |  |
| POST | Handles logging in the given user and returning their data to the sender |
| Logout |  |
| POST | Updates the given user’s logged in status in the database to false |
| SignUp |  |
| POST | Creates a new user in the database with the given data and approved status of false |

קישור לDB: <https://cloud.mongodb.com/v2/6769e9b50e05ac5c4ca41914#/metrics/replicaSet/6769ec2702953b376c0c22fd/explorer/art_market_db/artists/find>

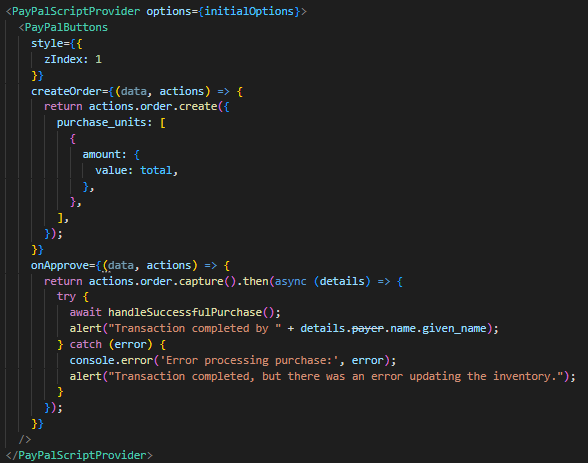
השתמשנו בחבילות mongodb וmongoose כדי לטפל בכל מה שקשור בתקשורת עם מסד הנתונים בMongoDB.



ובנוסף השתמשנו בPayPal Sandbox כדי לאפשר תשלום מאובטח ומדומה, הפונקציונליות ניתנה לנו ע"י התקנת @paypal/react-paypal-js ושימוש ב PayPalButtons וPayPalScriptProvider מתוכו.



<https://developer.paypal.com/tools/sandbox/>



פרומפטים לכלי AI:

“In a next.js project, how can I use mongoDB to upload and download documents?”

“Can you give me an example of how to use custom hooks?”

“What are models in terms of mongoDB and next.js? how do I use them to create objects and upload/download them to the database?”

6. יש להגיש תיק משתמש הכולל צילומי מסך והסברים כיצד להשתמש באתר שלכם.

פירוט המסכים:

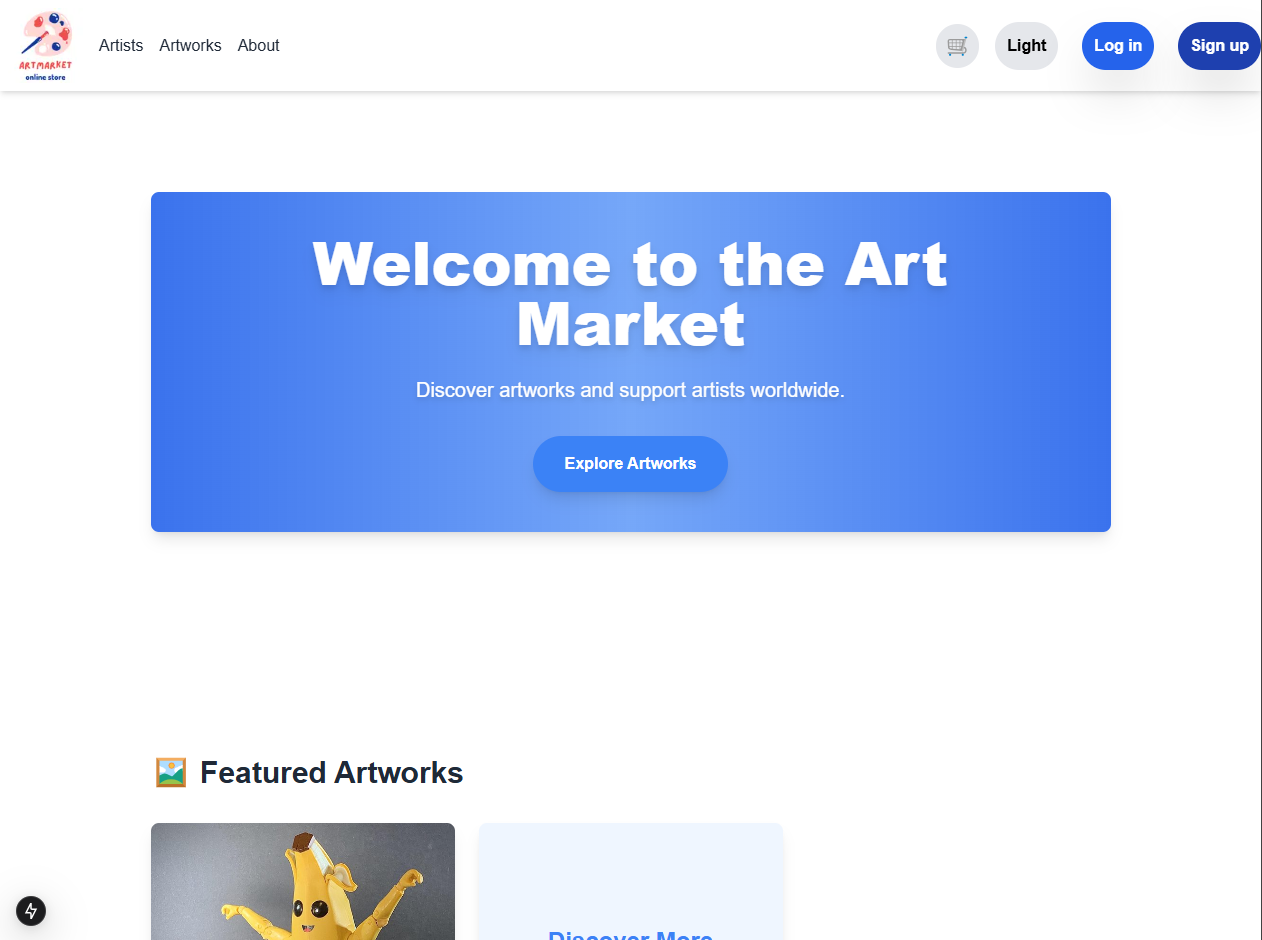
**מסך הבית:**

מסך הבית הוא הנקודה הראשונה שבה המשתמשים פוגשים את האתר. מטרתו להציג את התוכן המרכזי של האתר, להדגיש יצירות אמנות נבחרות, להציג אמנים מוצגים, ולהכווין את המשתמשים לגלות עוד יצירות ואמנים.

כיצד להשתמש במסך הבית?

1. גלילה ראשונית: המשתמש יכול לגלול מטה כדי להכיר את התוכן המרכזי של האתר.

1. לחיצה על יצירות אמנות: בלחיצה על כפתור "See All Artworks" ניתן לעבור לדף המציג את כל היצירות אמנות באתר ובלחיצה על היצירה תעביר את המשתמש לעמוד היצירה הספציפית שלה.
2. מעבר לאמנים נוספים: בלחיצה על כפתור "View All Artists" ניתן לעבור לדף המציג את כל האמנים באתר ובלחיצה על האמן תעביר את המשתמש לעמוד האמן הספציפי.
3. ניווט דרך הסרגל העליון: ניתן לעבור לדפים נוספים כמו עמוד אמנים, עמוד אומנות, אודות עגלה התחברות והרשמה.

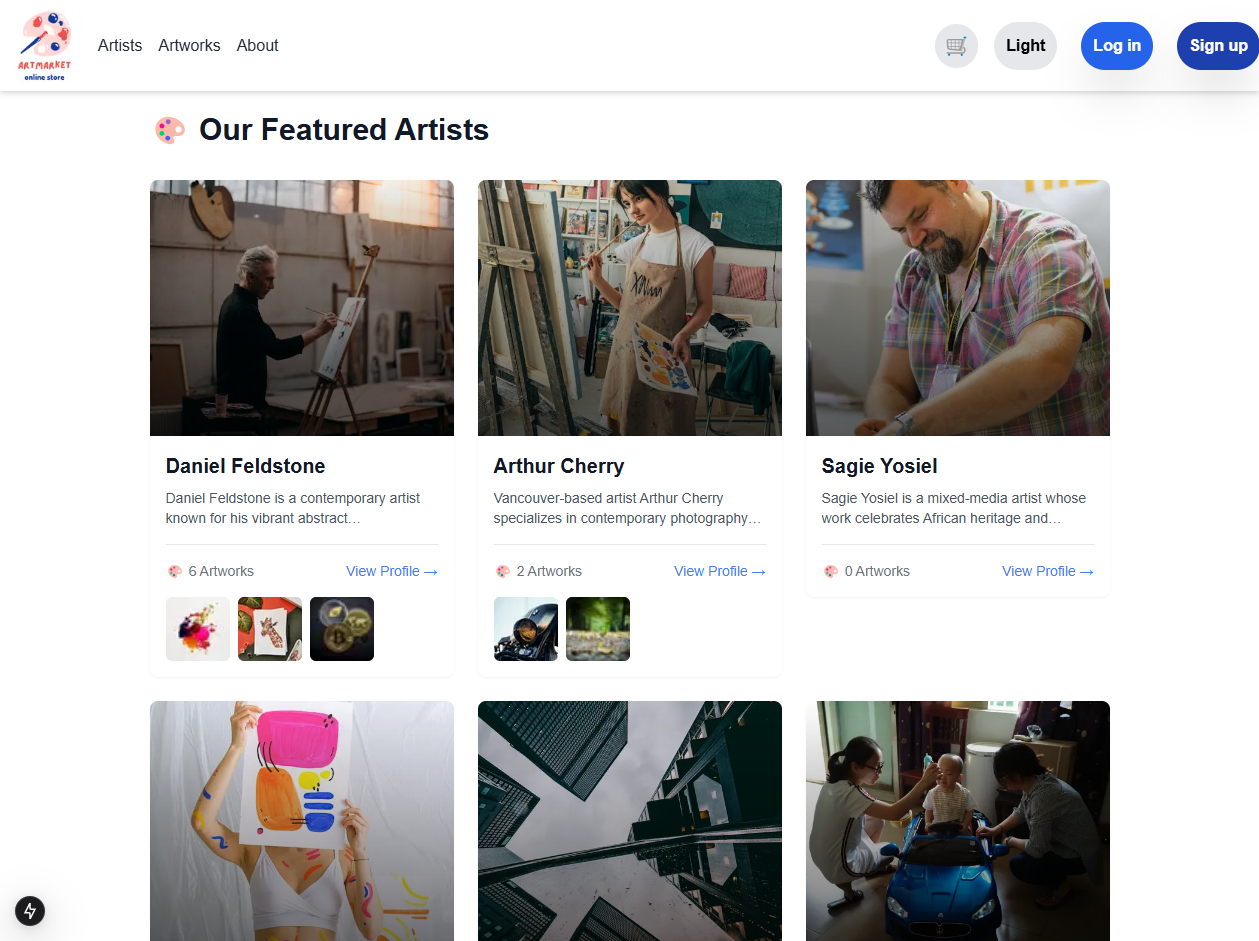


**מסך אמנים:**

מסך זה מציג את האמנים המשתתפים באתר, כולל מידע עליהם ויצירותיהם. המשתמש יכול לעיין ברשימת האמנים, לבחור אמן מסוים ולראות את יצירותיו.

כיצד להשתמש במסך זה?

1. דפדוף בין אמנים: המשתמש יכול לגלול ולראות את רשימת האמנים הקיימים באתר.
2. בחירת אמן: לחיצה על אמן מסוים תציג את פרטיו ויצירותיו.

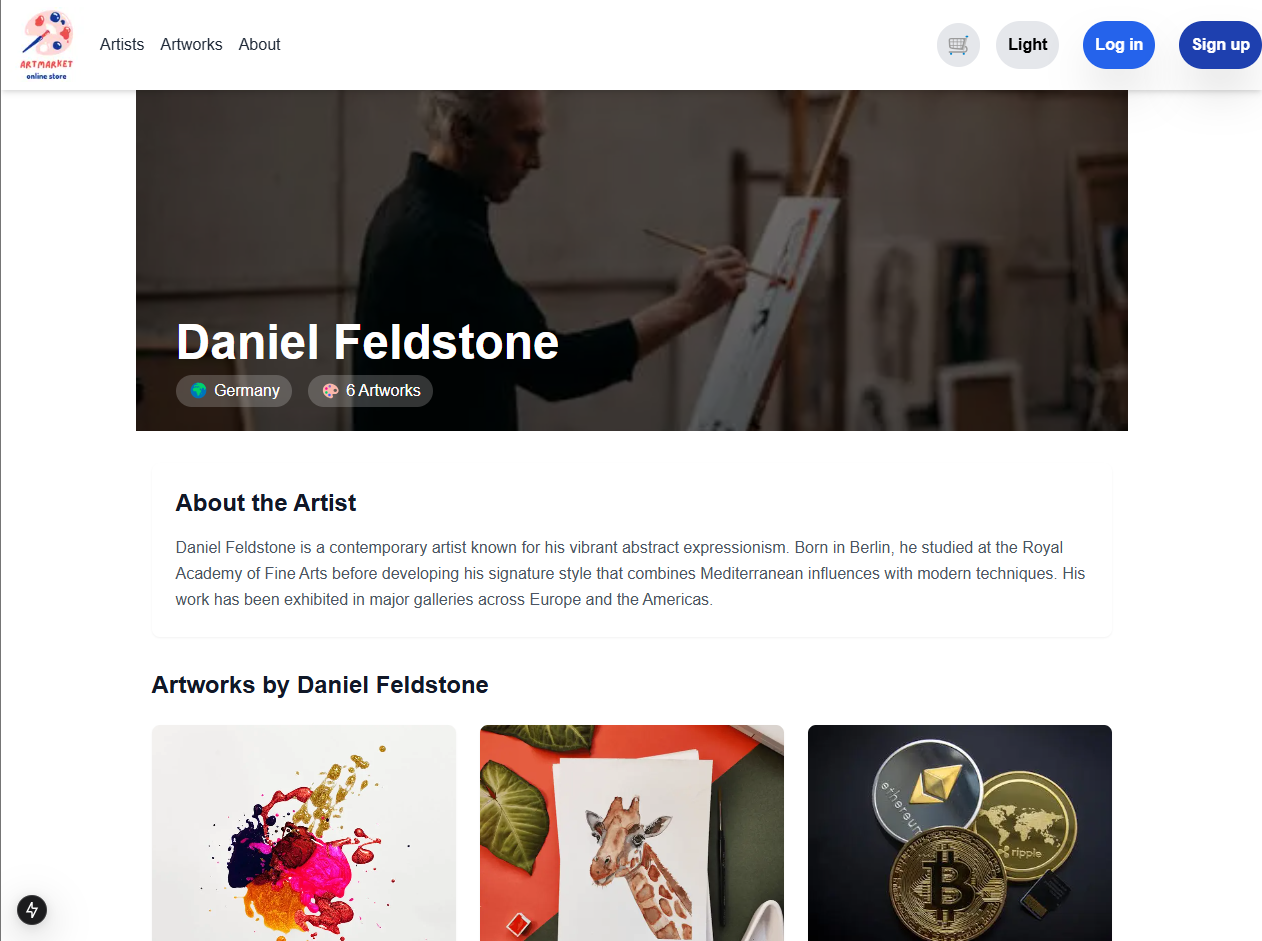


**מסך פרופיל אמן ציבורי:**

מסך זה מציג את הפרופיל הציבורי של אמן באתר. כאן משתמשים יכולים ללמוד על האמן, לראות את היצירות שלו, ולחזור חזרה לרשימת האמנים. המסך כולל תמונת רקע, ביוגרפיה, ורשימת היצירות של האמן.

כיצד להשתמש במסך זה?

1. כניסה לפרופיל האמן: המשתמש מגיע לעמוד דרך רשימת האמנים או דרך קישור ישיר לפרופיל.
2. קריאת מידע על האמן: ניתן לראות את שם האמן, המדינה שלו, ואת הביוגרפיה שלו.
3. צפייה ביצירות האמן: ניתן לעיין ביצירות האמן, כאשר כל יצירה מוצגת בצורה בולטת.
4. חזרה לרשימת האמנים: לחיצה על הכפתור "Back to Artists" תחזיר את המשתמש לרשימת האמנים.

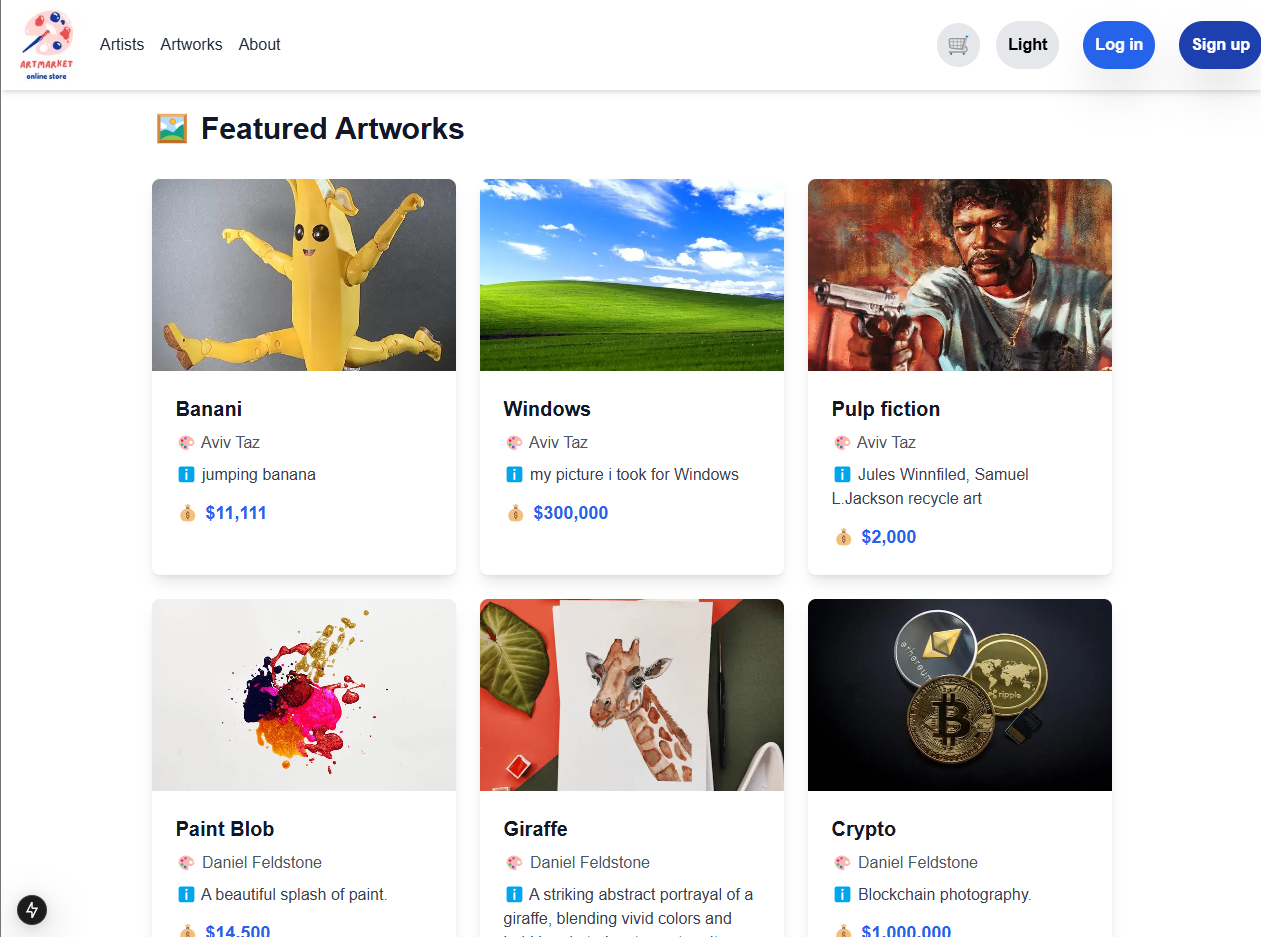


**מסך יצירות אמנות:**

מסך זה מציג את כל היצירות הקיימות באתר. המשתמש יכול לעיין ביצירות, לבחור יצירה מסוימת ולעבור לעמוד הייעודי שלה.

כיצד להשתמש במסך זה?

1. סקירת יצירות: המשתמש יכול לגלול ולראות את כל היצירות הקיימות.
2. מעבר לעמוד יצירה: בלחיצה על יצירה, המשתמש יועבר לעמוד היצירה הספציפית שלה.

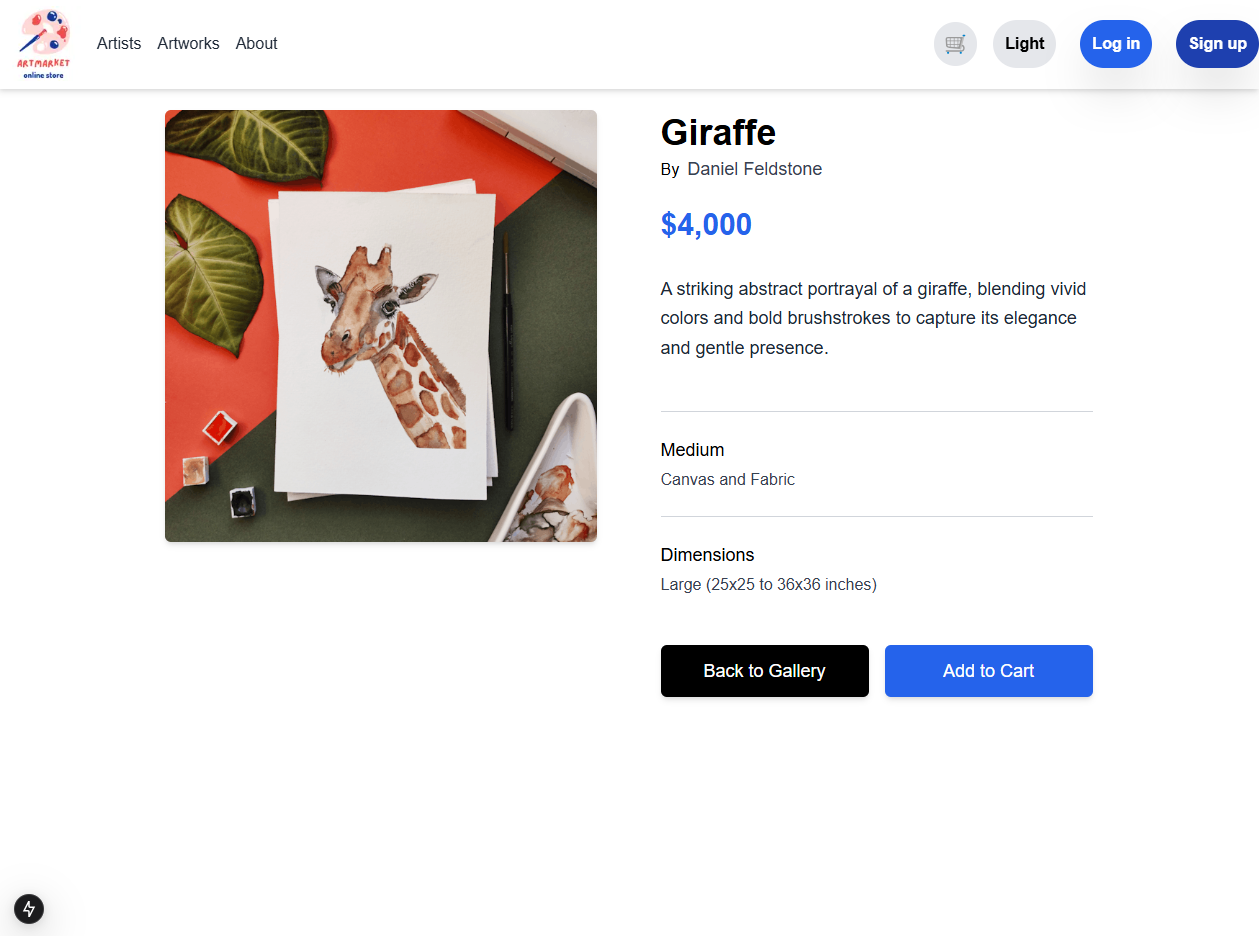


**מסך יצירת אמנות:**

מסך זה מציג דף מידע על יצירה מסוימת. כאן ניתן לראות את פרטי היצירה, את האמן שיצר אותה, את מחירה, תיאור היצירה, וכן להוסיף אותה לעגלת הקניות או לחזור לגלריה.

כיצד להשתמש במסך זה?

1. עיון בפרטי היצירה: המשתמש יכול לראות תמונה, שם, תיאור ופרטים טכניים של היצירה.
2. מעבר לעמוד האמן: לחיצה על שם האמן תעביר את המשתמש לעמוד הפרופיל הציבורי של האמן.
3. רכישת היצירה: לחיצה על "Add to Cart" תוסיף את היצירה לעגלת הקניות.
4. חזרה לגלריה: ניתן ללחוץ על "Back to Gallery" כדי לחזור לדף כל היצירות.

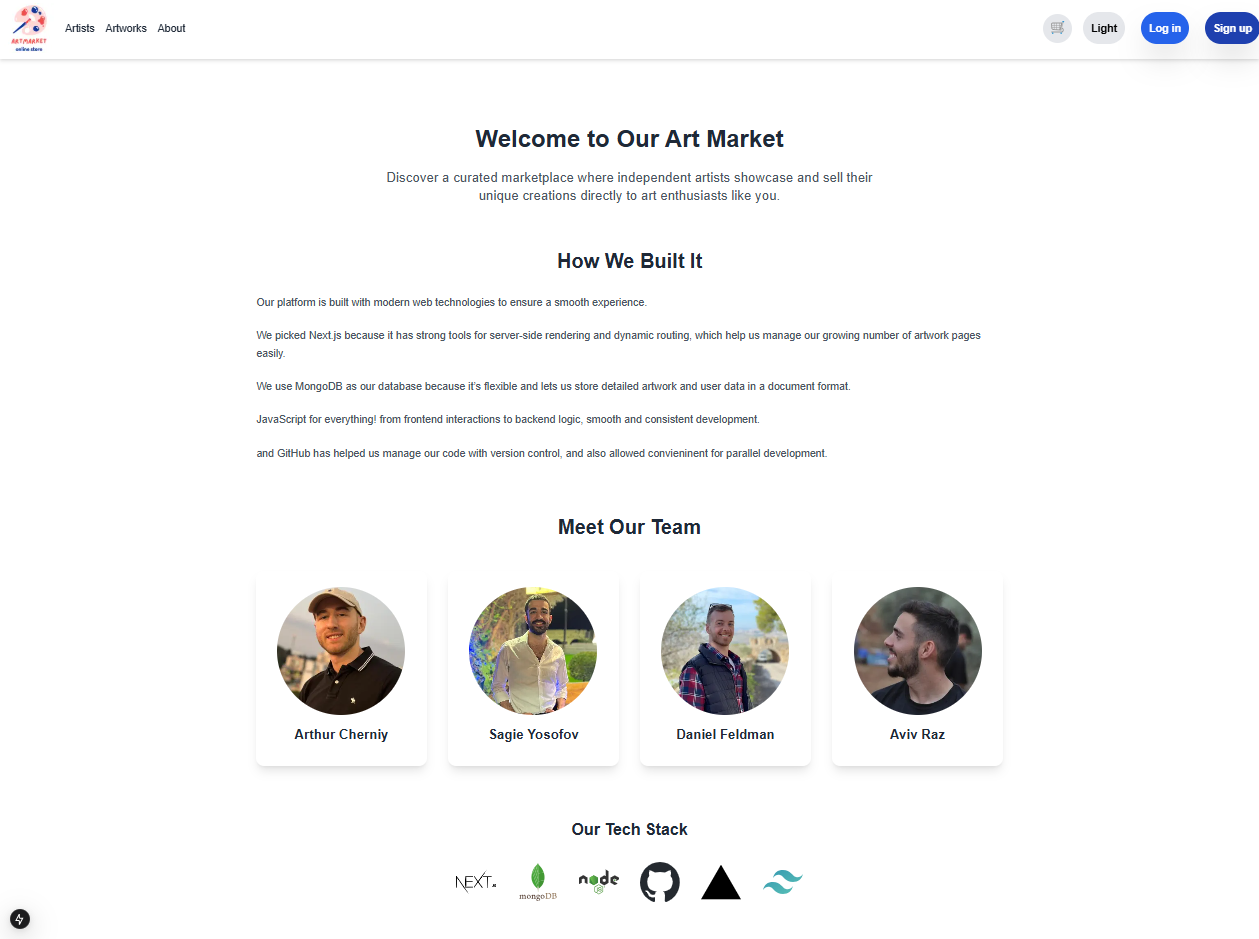


**מסך אודות (About):**

מסך זה מציג מידע על הפלטפורמה, כולל הרעיון מאחוריה, הטכנולוגיות ששימשו לפיתוחה, הצוות שעומד מאחוריה, והזמנה למשתמשים להתחיל לגלות יצירות אמנות.

כיצד להשתמש במסך זה?

1. לקרוא על הרקע של הפלטפורמה: המשתמש יכול להבין כיצד נבנה האתר ואילו טכנולוגיות משמשות אותו.
2. להכיר את הצוות: ניתן לעיין בפרטים על הצוות שעומד מאחורי הפרויקט, לחיצה על חבר הצוות תוביל לפרופיל הלינקדין שלו.
3. להתחיל לגלוש ביצירות אמנות: לחיצה על כפתור "Start Exploring" תיקח את המשתמש לעמוד יצירות האמנות.

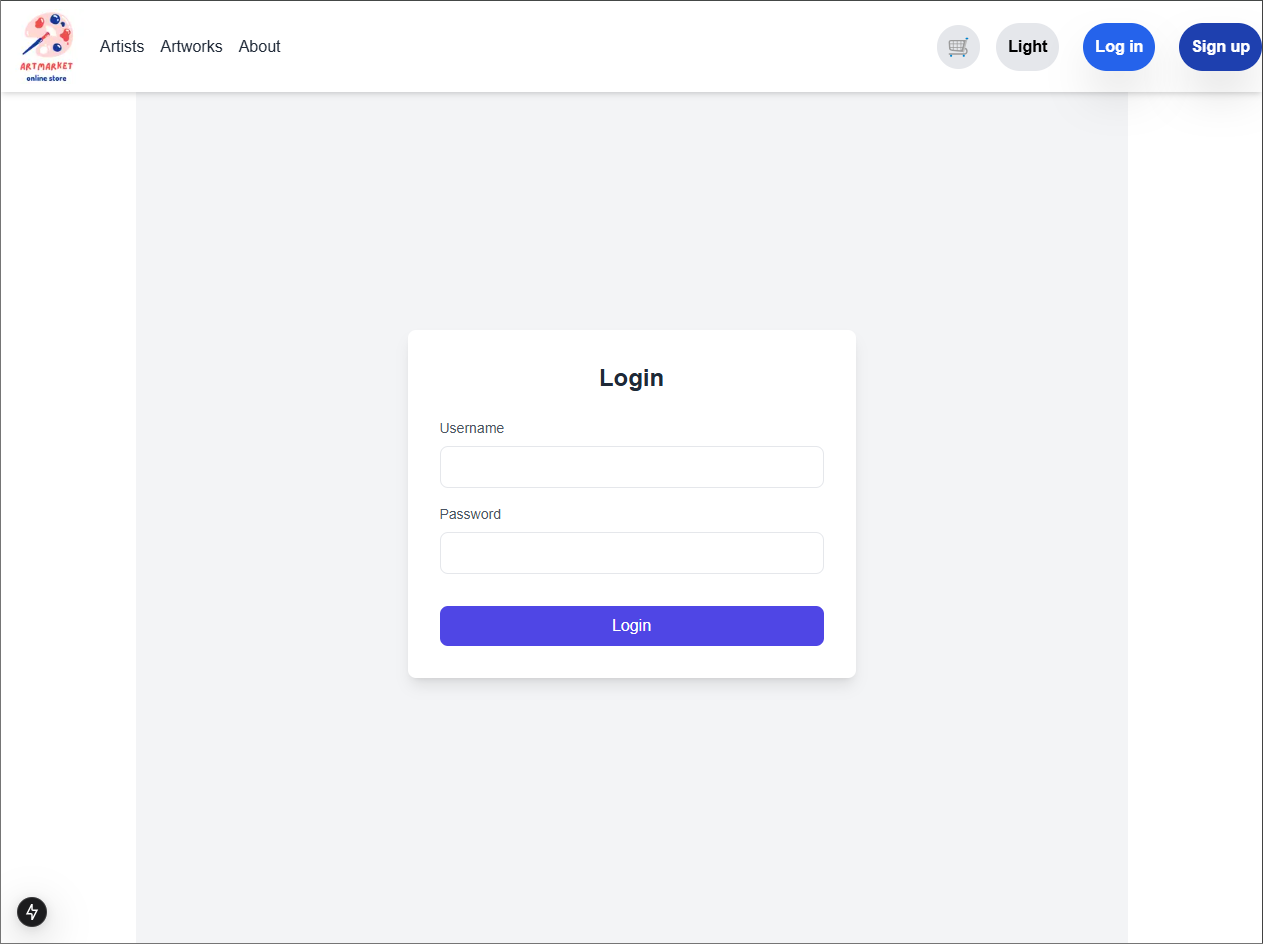


**מסך התחברות:**

מסך זה מאפשר למשתמשים להתחבר למערכת באמצעות שם משתמש וסיסמה. העיצוב ידידותי, עם דגש על חוויית משתמש חלקה ואינטראקטיבית.

כיצד להשתמש במסך זה?

1. הזנת שם משתמש וסיסמה: המשתמש ממלא את שדות ה-Username וה-Password.
2. לחיצה על כפתור "Login": שליחת הנתונים לבדיקה.
3. קבלת הודעה בהתאם למצב:
   1. אם הנתונים נכונים → המשתמש עובר לעמוד המתאים.
   2. אם הנתונים שגויים → מתקבלת הודעה באדום.

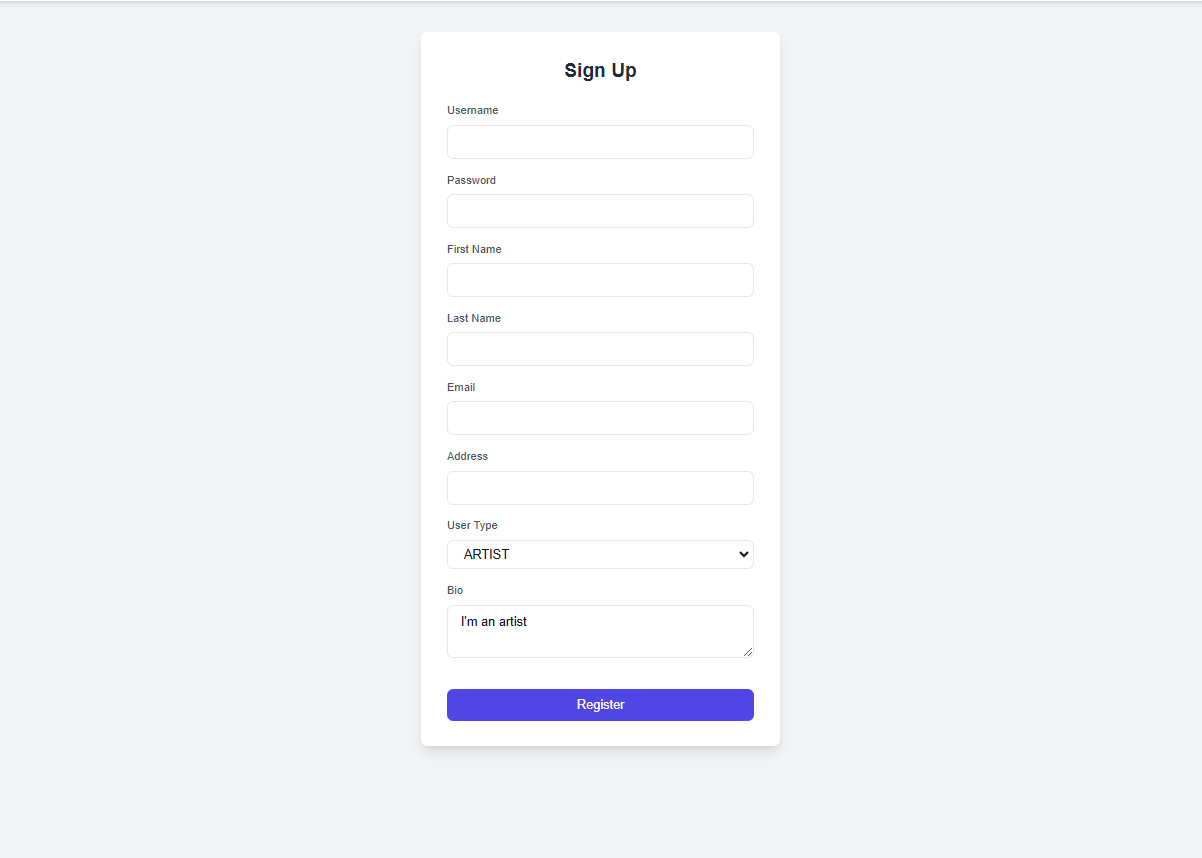


**מסך הרשמה:**

מסך זה מאפשר למשתמשים חדשים להירשם לאתר על ידי הזנת פרטים אישיים, בחירת סוג משתמש (Admin או Artist), ומילוי ביוגרפיה.

כיצד להשתמש במסך זה?

1. מילוי הפרטים האישיים: יש להזין שם משתמש, סיסמה, שם מלא, מייל וכתובת.
2. בחירת סוג משתמש: אם מדובר באמן (ARTIST), יש למלא גם ביוגרפיה קצרה.
3. לחיצה על "Register": הנתונים יישלחו לבדיקה והרשמה לשרת.
4. קבלת משוב על ההרשמה:
   1. אם ההרשמה הצליחה → הודעת אישור.
   2. אם יש בעיה → הודעת שגיאה.

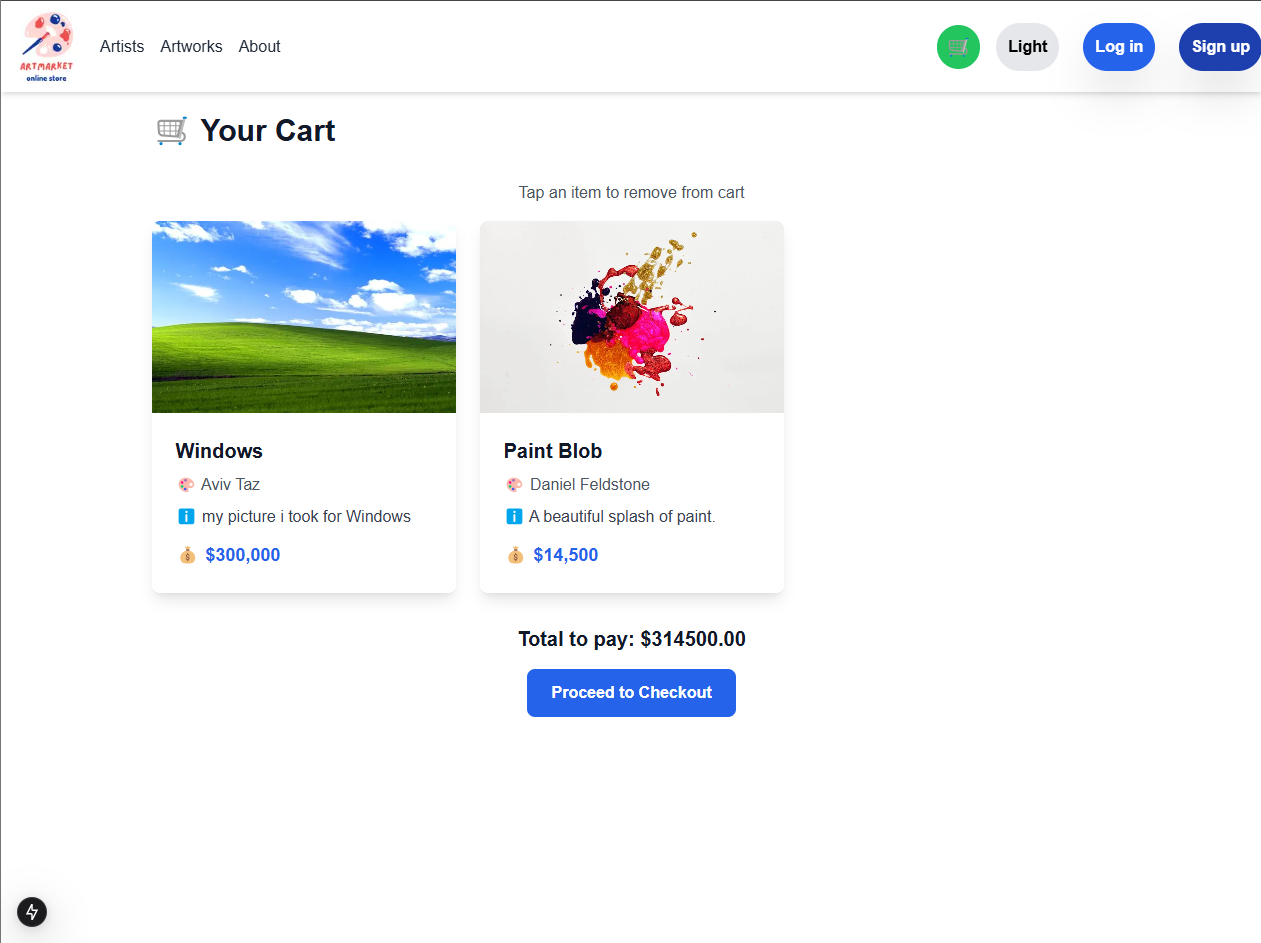


**מסך עגלה:**

מסך זה מציג את פריטי האמנות שהמשתמש הוסיף לעגלת הקניות. המשתמש יכול להסיר פריטים מהעגלה ולבצע רכישה.

כיצד להשתמש במסך זה?

1. סקירת פריטים בעגלה: ניתן לראות את כל היצירות שנוספו לעגלה.
2. הסרת פריטים מהעגלה: לחיצה על יצירה תסיר אותה מהעגלה.
3. מעבר לתשלום: לחיצה על "Proceed to Checkout" תעביר את המשתמש לתהליך הרכישה.

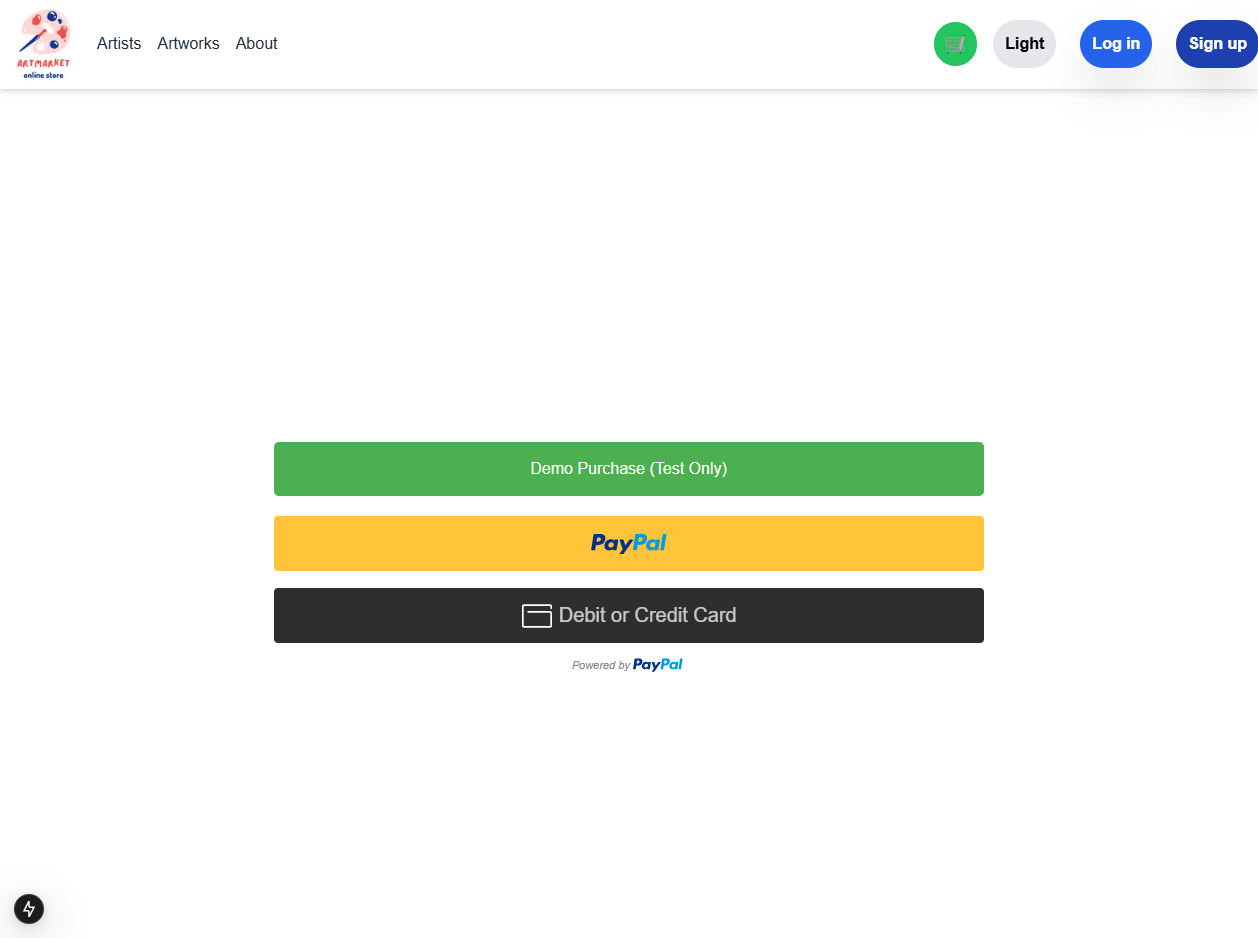


**מסך תשלום:**

מסך זה מאפשר למשתמשים להשליםאת הרכישה באמצעות PayPal או רכישה דמו (למטרות בדיקה). המשתמש יכול לראות את הסכום הכולל לתשלום ולבצע תשלום מאובטח.

כיצד להשתמש במסך זה?

1. בחירת אמצעי תשלום
   1. לחיצה על "Demo Purchase" לבדיקת התהליך ללא תשלום אמיתי.
   2. לחיצה על כפתור **PayPal** לתשלום דרך פייפאל.
   3. לחיצה על כפתור Debit or Credit Card לתשלום בכרטיס אשראי.

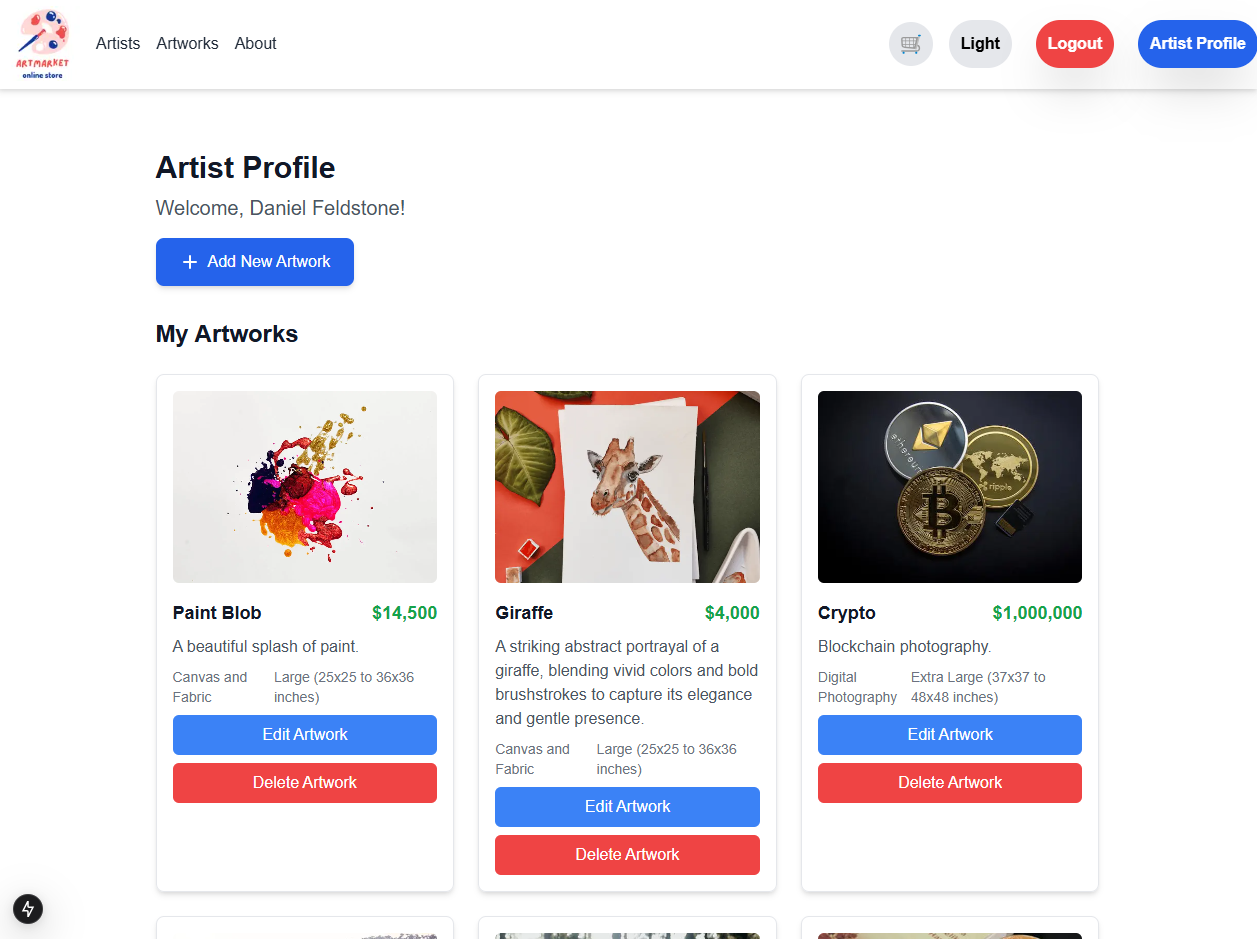


**מסך פרופיל אמן אישי:**

מסך זה מאפשר לאמן לראות ולנהל את פרופיל האמן האישי שלו. כאן ניתן לצפות ביצירות שהועלו, להוסיף יצירות חדשות, ולנהל את הגלריה האישית.

כיצד להשתמש במסך זה?

1. הוספת יצירה חדשה:
   1. לחיצה על "Add New Artwork" תפתח טופס להוספת יצירה.
   2. ניתן לבטל את הפעולה באמצעות לחיצה על "Cancel" .
2. צפייה ביצירות קיימות
   1. היצירות של המשתמש יוצגו ברשימה נוחה למעבר.
   2. אם אין יצירות, המשתמש יקבל הצעה להוסיף יצירה ראשונה.

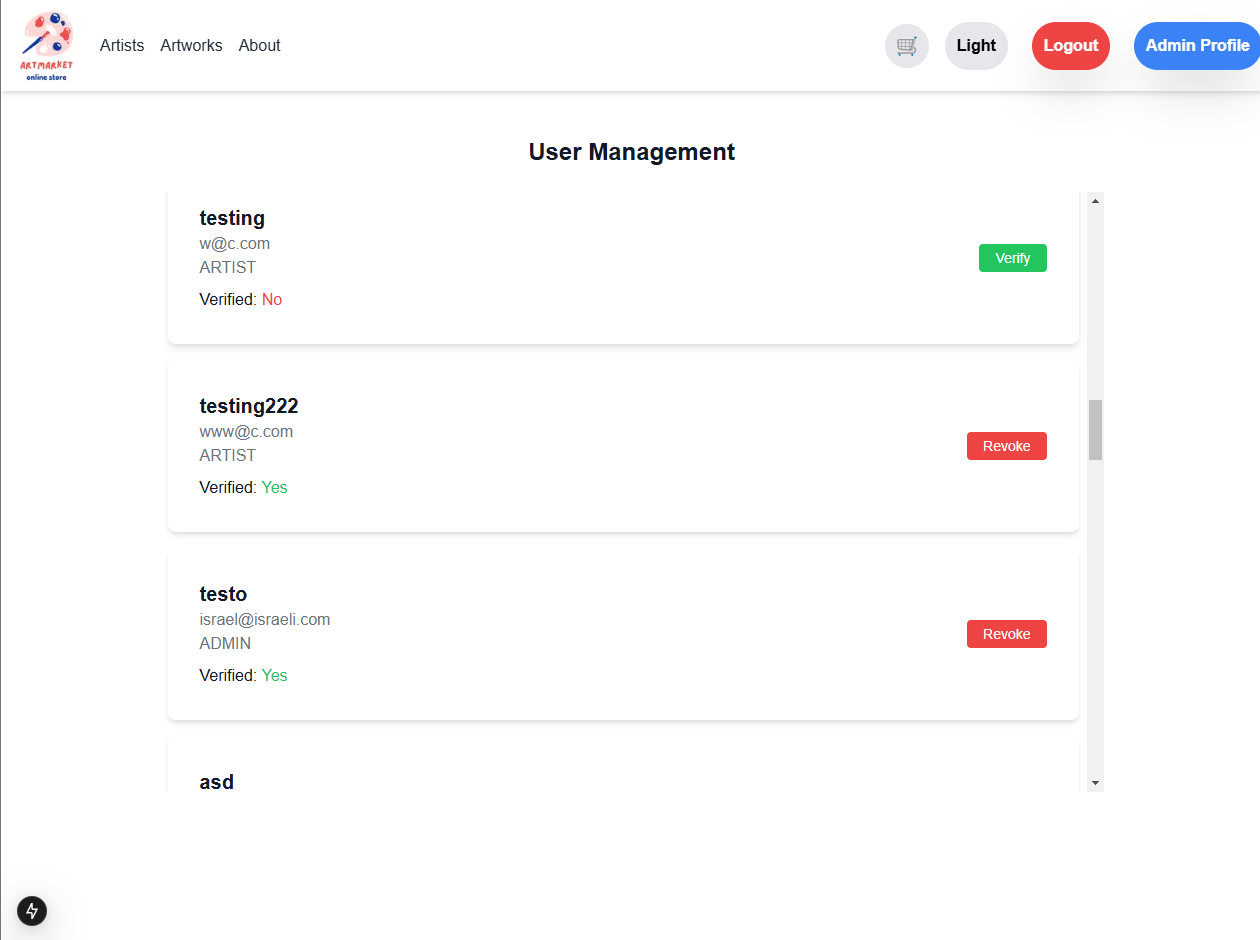


**מסך אדמין:**

מסך זה מאפשר למנהלי המערכת לנהל את המשתמשים הרשומים באתר. ניתן לצפות ברשימת המשתמשים, לראות את פרטיהם ולשנות את סטטוס האימות שלהם (מאומת/לא מאומת).

כיצד להשתמש במסך זה?

1. סקירת המשתמשים הרשומים: ניתן לראות את כל המשתמשים הרשומים באתר (למעט המשתמש המחובר).
2. אימות משתמשים:
   1. לחיצה על "Verify" מאמתת את המשתמש.
   2. לחיצה על "Revoke" מבטלת את האימות של המשתמש.



7. בתאריכים 26-27.1 תציגו את תוצרי הפרויקט שלכם. כל צוות יציג 15 דקות את המערכת, וכן אלמנטים מרכזיים של העבודה (סעיפים 1-4 של מסמך זה).בסיום ההצגה נשאל את חברי הצוות שאלות בנוגע להצגה ולנושאי הקורס.

מטרת ההצגה היא לתת משוב ומיקוד לקראת הצגת הפרויקט.

יש להשתבץ להצגה בקישור שפורסם לכם, חובת ההצגה על כל חברי הצוות!

להזכירכם – ציון אישי של 10% יינתן לכל אחד מהמציגים לפי הצגה ומענה על שאלות.

יש להגיש את כל התוצרים :

* GIt repository - קישור (הריפו שפתחתם לצוות)
* Git Pages - קישור
* **הגשה בתיבת ההגשה במוודל – "הגשת פרויקט**"- נא להגיש שם את קובץ הפרויקט הכולל מענה לשאלות 1-6

בנוסף, יש לוודא כי תיקיית ה- GIT ציבורית וכוללת את כל הקבצים של הפרויקט, כולל המסמכים, המצגת וכן הוראות הרצה.

הנחיות הגשה:

1.באחריותכם לוודא שהגשתם את כל התוצרים כנדרש ושהם הגיעו ליעדם.

לא תתאפשר בדיקה מחודשת של העבודה עקב טעויות בעת ההגשה!

2 .יש להגיש את התרגיל בקבוצות שהוגדרו בקורס.

3 .יש להגיש את התוצר הסופי של מסמך הפרויקט בפורמט WORD שייכלל בריפו של הפרויקט.

4.ניתן להפנות שאלות במייל לצוות הקורס, נא לשלוח שאלות לכלל הצוות . נא לציין את שם הקורס.

**שימו לב – יש לכלול בתחילת העבודה דף שער כמפורט למטה:**

# יש לכלול בתחילת מסמך הפרויקט את דף השער הבא: **טכנולוגיות אינטרנט מתקדמות - 61776 (WEB)**

**הגשת פרויקט**

**<שם הפרויקט> < קוד הפרויקט><מס קבוצה>**

|  |  |
| --- | --- |
| **שם חבר.ת הצוות** | **תז** |
|  |  |
|  |  |
|  |  |
|  |  |

<תקציר הפרויקט - עד חצי עמוד- פונקציונליות מרכזיות לפי משתמשים>

< מימוש-שמות הטכנולוגיות המרכזיות בכל אחד מהחלקים -/styling/db/ Backend/frontend>

<קישור לתיקיית גיט ציבורי>

<קישור לאתר>