Web development final project

Oriana Eid (211113)

William Abou Khalil (210241)

24/11/2023

FusArt, the Fusion of Art

Introduction

Artists have always struggled to live from their art, especially in the Middle East, and even more these days with the rise of AI. It's not been used as a tool anymore, but as a replacement for human talents and capacities. Furthermore, the data used to feed these AI models and never credited for the original owners. We can see that from websites specialized in generating art pieces from a search engine.

Problematic

How can artists make a living out of their work in the Middle East?

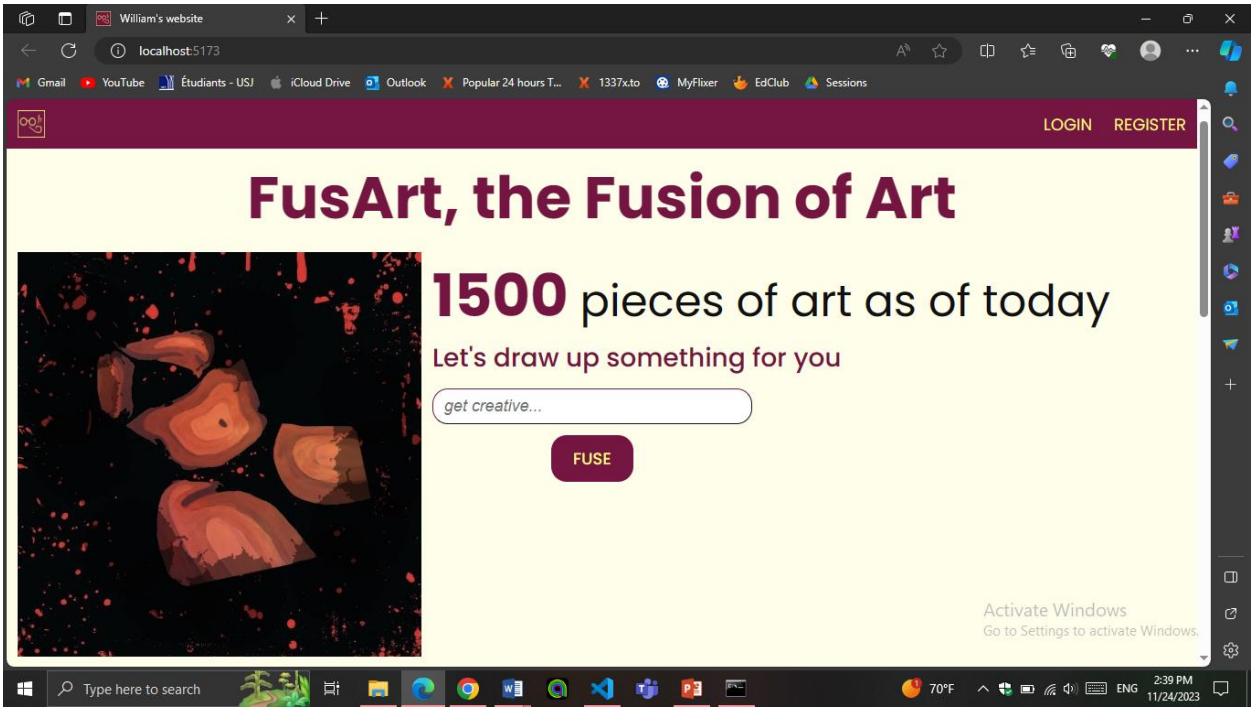
How can we fight the domination of AI in art?

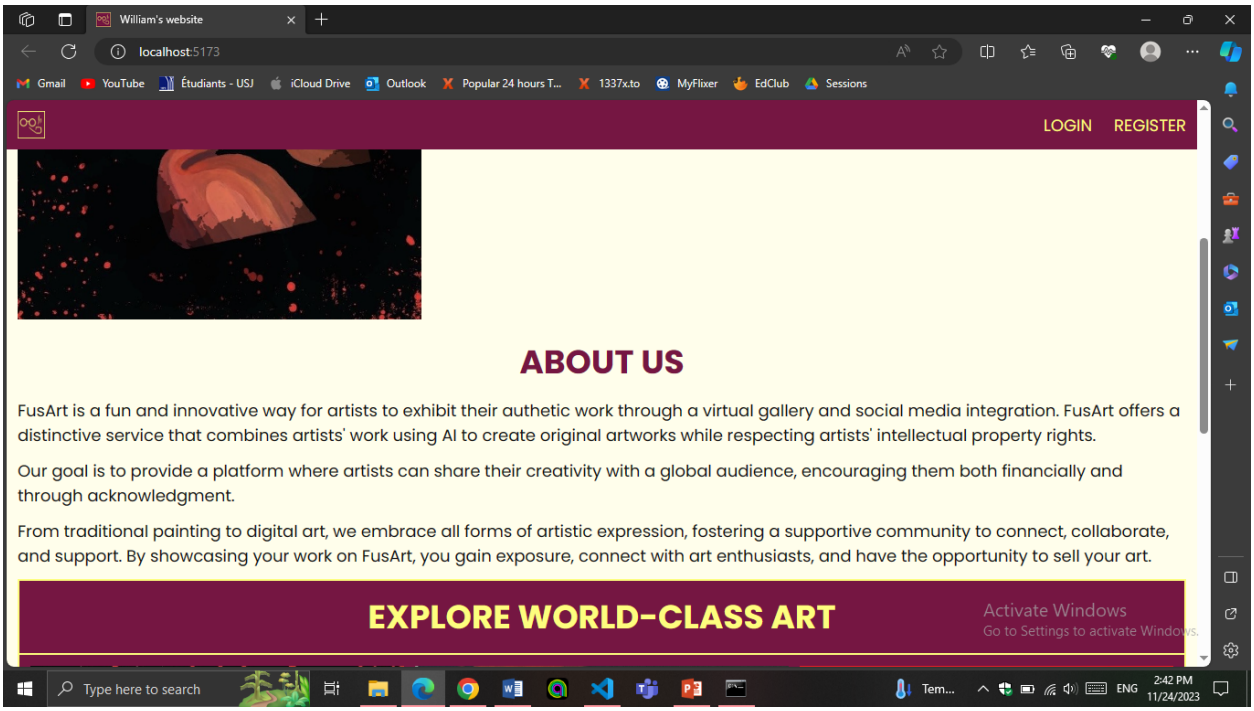
Solution

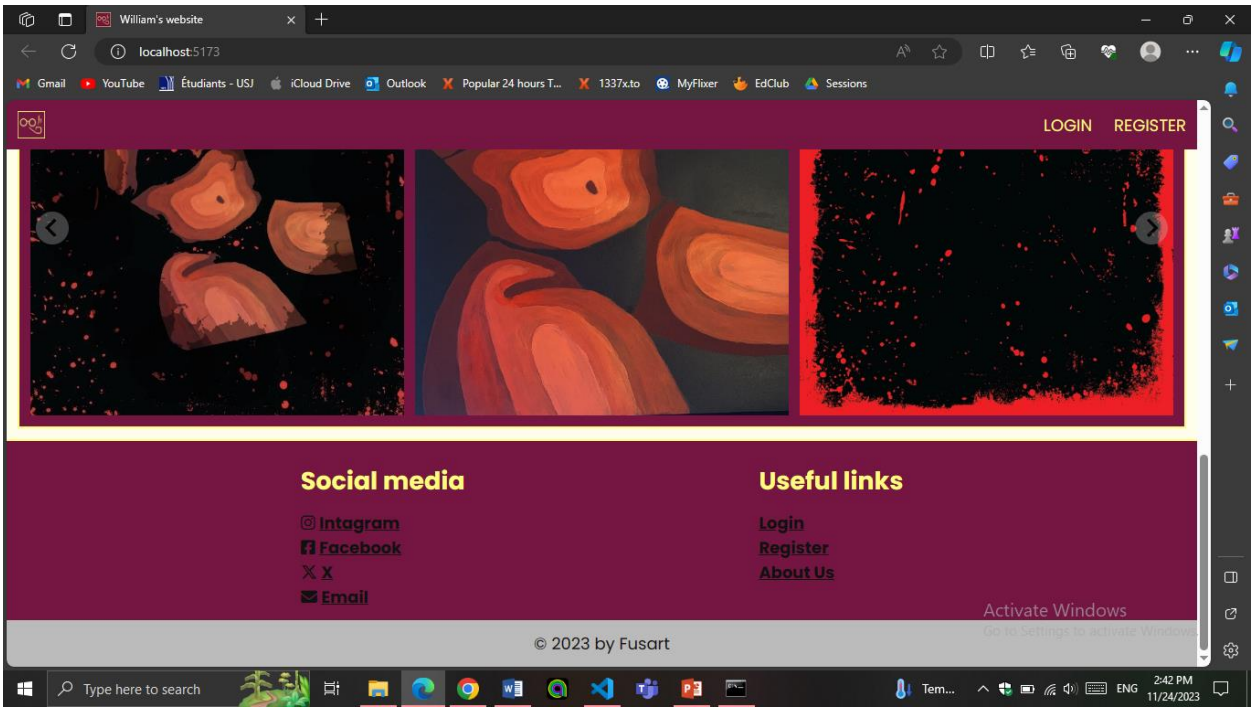
FusArt is a website with the aim of providing a platform where artists can share their creativity with a global audience, encouraging them both financially and by preserving their name. It is a fun and innovative way for artists to showcase their authentic work through a virtual gallery and integration with social media. All forms of artistic expression are encouraged, whether traditional painting or digital art, fostering a community to gain visibility, connect with art enthusiasts, and have the opportunity to sell art.

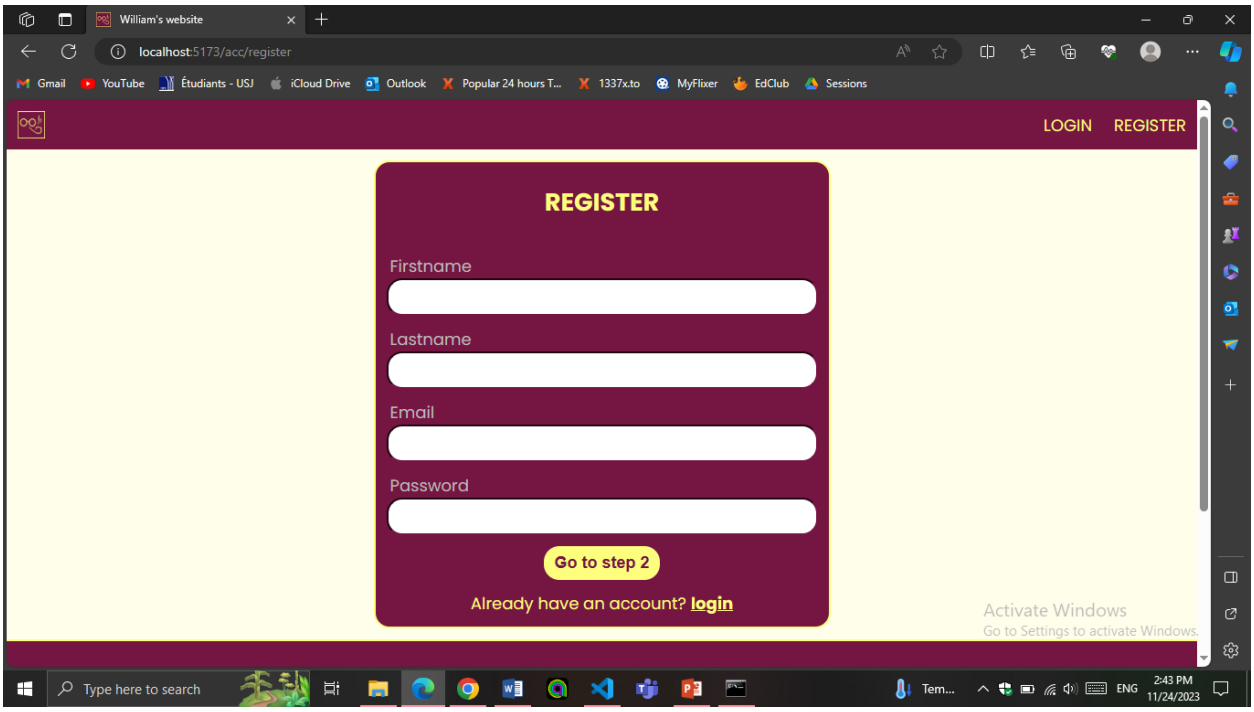
FusArt offers a distinctive service that combines the work of artists using AI to create original pieces while respecting the intellectual property rights of artists, unlike what is common nowadays regarding the use of AI in art. Indeed, the AI model used does not generate art pieces using hidden and unknown user data but combines parts of art pieces stored in our database that artists publish on the website with the goal of earning money and being recognized.

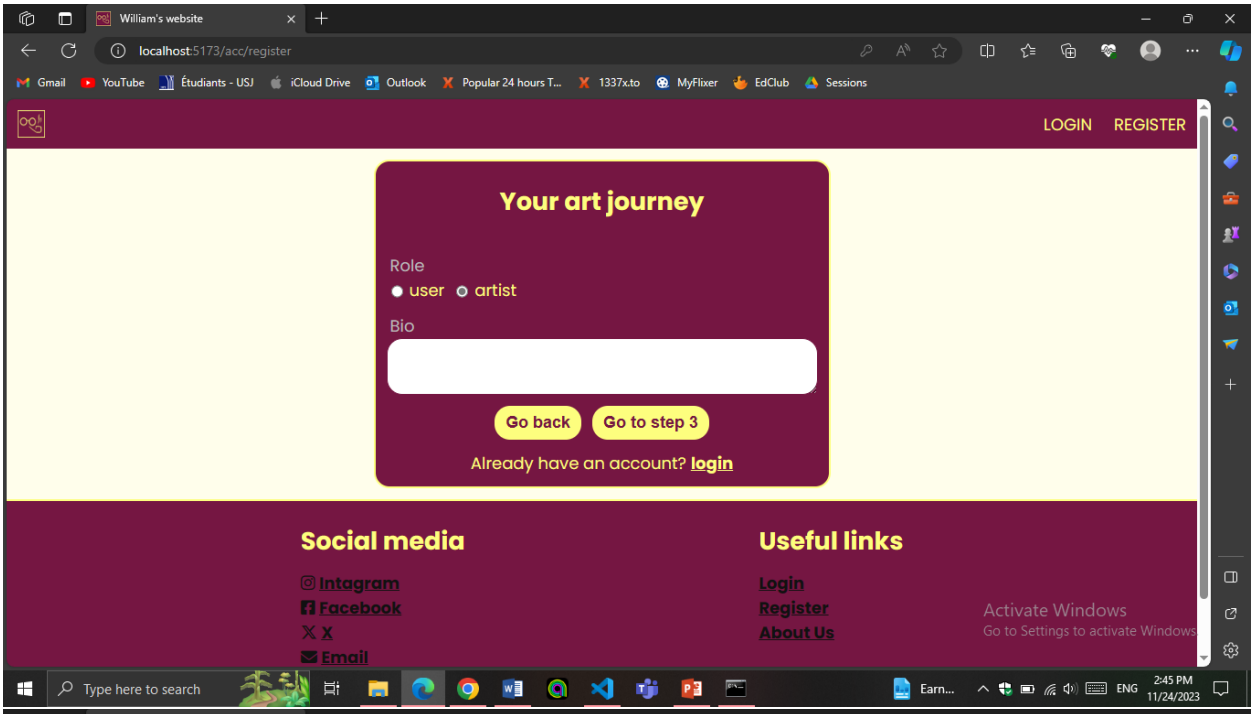
Screens

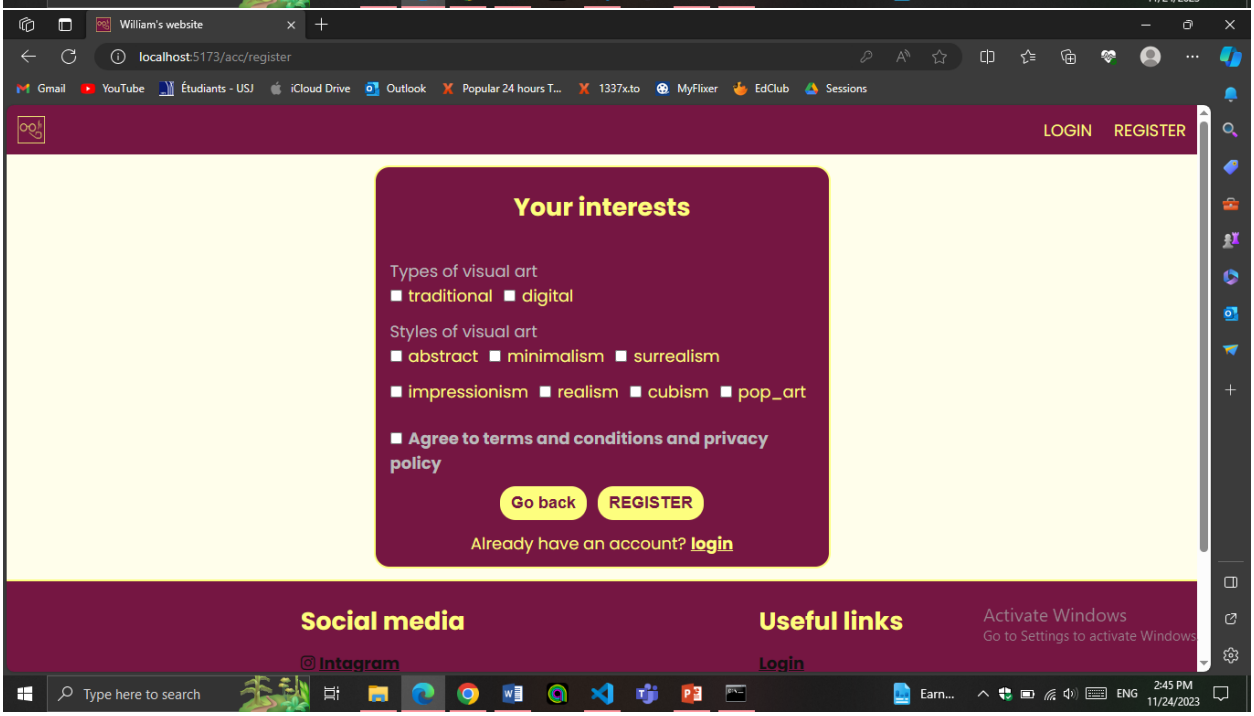

William's website
localhost:5173
LOGIN
REGISTER
FusArt, the Fusion of Art
1500 pieces of art as of today
Let's draw up something for you
get creative...
FUSE
Activate Windows
Go to Settings to activate Windows.

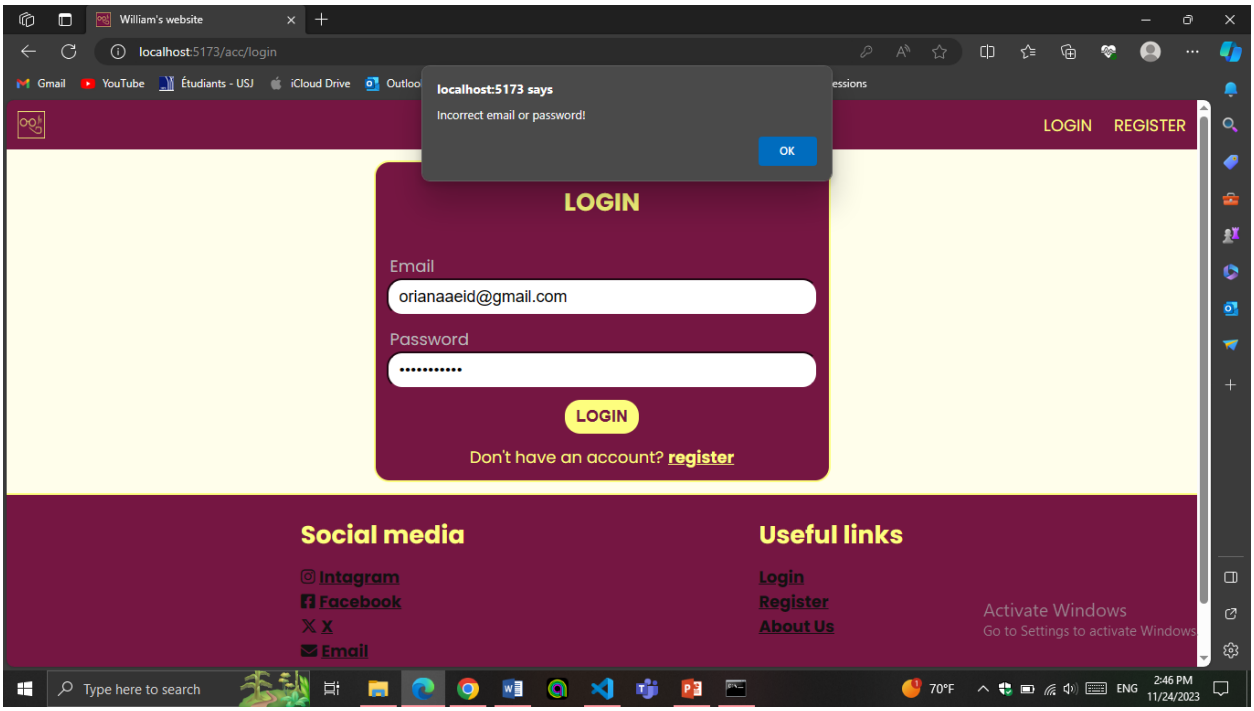

William's website
localhost:5173
LOGIN
REGISTER
ABOUT US
FusArt is a fun and innovative way for artists to exhibit their authetic work through a virtual gallery and social media integration. FusArt offers a distinctive service that combines artists' work using AI to create original artworks while respecting artists' intellectual property rights.
Our goal is to provide a platform where artists can share their creativity with a global audience, encouraging them both financially and through acknowledgment.
From traditional painting to digital art, we embrace all forms of artistic expression, fostering a supportive community to connect, collaborate, and support. By showcasing your work on FusArt, you gain exposure, connect with art enthusiasts, and have the opportunity to sell your art.
EXPLORE WORLD-CLASS ART
Activate Windows
Go to Settings to activate Windows.

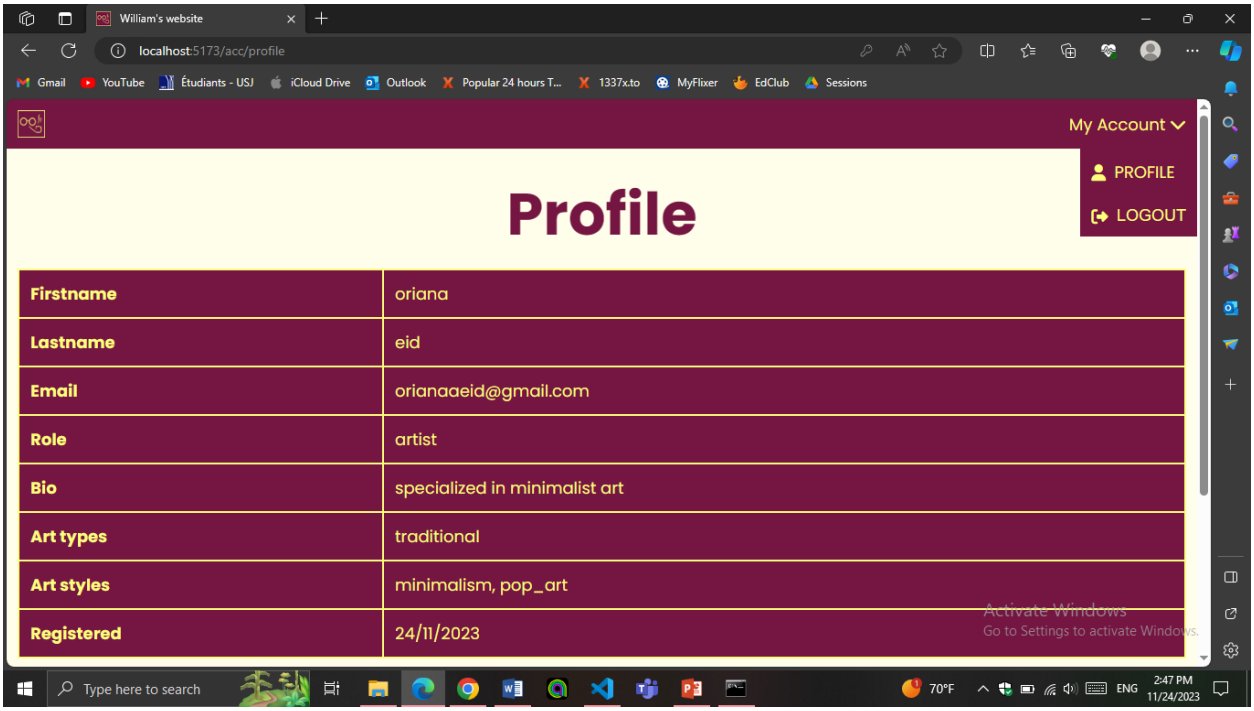

William's website
localhost:5173
LOGIN
REGISTER
Social media
Intagram
Facebook
X
Email
Useful links
Login
Register
About Us
© 2023 by Fusart


William's website
localhost:5173/acc/register
LOGIN
REGISTER
REGISTER
Firstname
Lastname
Email
Password
Go to step 2
Already have an account? login


William's website
localhost:5173/acc/register
LOGIN
REGISTER
Your art journey
Role
user
artist
Bio
Go back
Go to step 3
Already have an account? login
Social media
Intagram
Facebook
X
Email
Useful links
Login
Register
About Us
Activate Windows
Go to Settings to activate Windows


William's website
localhost:5173/acc/register
LOGIN
REGISTER
Your interests
Types of visual art
traditional
digital
Styles of visual art
abstract
minimalism
surrealism
impressionism
realism
cubism
pop_art
Agree to terms and conditions and privacy policy
Go back
REGISTER
Already have an account? login
Social media
Intagram
Useful links
Login
Activate Windows
Go to Settings to activate Windows


William's website
localhost:5173/acc/login
localhost:5173 says
Incorrect email or password!
OK
LOGIN
REGISTER
LOGIN
Email
orianaaeid@gmail.com
Password
LOGIN
Don't have an account? register
Social media
Intagram
Facebook
X
Email
Useful links
Login
Register
About Us
Activate Windows
Go to Settings to activate Windows.
Type here to search
70°F
ENG
2:46 PM
11/24/2023


William's website
localhost:5173/acc/profile
My Account
PROFILE
LOGOUT
Profile
Firstname
oriana
Lastname
eid
Email
orianaaeid@gmail.com
Role
artist
Bio
specialized in minimalist art
Art types
traditional
Art styles
minimalism, pop_art
Registered
24/11/2023
Activate Windows
Go to Settings to activate Windows.
Type here to search
70°F
ENG
2:47 PM
11/24/2023

Process

1. Server.js setup: very basic backend for our API
2. API routes: folder routers, user.js. 3 post requests. It was a bit hard to configure due to not practically knowing how to encrypt and authenticate our data. We discovered bcrypt API to hash our data and save them in our database. We set up one post request for login, one for register and the third for info with error handling.
3. MongoDB Atlas and Mongoose (models and schemas): folder models, UserModel.js. We used the Mongoose library to set up our schemas that will be saved that way in our MongoDB Atlas database. We struggled to link MongoDB to our API because of network issues (IP addresses). We set up the db.config.js file with 3 variables: user, pass and cluster, so we can be more flexible in case we want to add more data or change the account.
4. Controllers: folder routers. Following the MVC model, the user.js is responsible for the control of the requests. We hardcoded all the possible scenarios while logging in and registering using if conditions. This could have been done using outsourcing
5. React App: App.jsx. We used jsx so we can implement HTML with JavaSript. Here, the front end is coded, it is the View in MVC. We added in the end the CSS style. It links the pages.
6. Pages and components: we hardcoded all the components and pages to get the best base for React apps.

npm packages

- `express` is used to simplify working with nodejs to build server APIs that occasionally (not always) interract with databases

- `mongoose` is used to simplify working with mongo database to manage collections, store, read, update and delete data with ease.

- `bcrypt` is used to hash data and compare them to non-hashed data, that way we add a layer of security to especially passwords in databases.

- `cors` is a cross-origin-resource-security package that allows for easy setup for CORS security options like allowed methods, domains, cookies, and so.

files and folders

1. [node_modules] stores the npm packages we download.

2. [db.config.js] has the basic configuration of mongodb, like to connect to it to be able to use its functions. we import this file in the [server.js] to establish the connection on API start.

3. [models] is a folder that stores the database configurations of every collection inside mongoDB.

4. [routers] is a folder that stores the route handlers (it means specific paths in the URL after the domain name)

e.g.: /user part is handled with [routers/user.js].

5. [server.js] is the manager of the API:

- we initialize the API,
- check with cors if the client is allowed access to the API,
- connect to the database,
- use middlewares such as express.json() to parse the incoming request's json data and put them inside `request.body` object,
- and finally, we listen to the port.

<u>Components</u>

- About: about us text in the home page
- Copyright
- Footer: social media, login, sign up
- HomeEntry: home page with images and search engine
- LoginStruct
- Navbar: navigation bar, logo, drop down menu
- PhotoSlider: display images
- RegisterStepThree: user/artist specifies the kind and type of art they're interested in
- RegisterStepTwo: chooses if user/artist and can write an optional biography

About:

```
export default function About() {
  return (
    <article id="about" style={{ marginBottom: "12px"}}>
      <h2 className="is-title">about us</h2>
      <section className="divide-y-3">
        <p>
          FusArt is a fun and innovative way for artists to exhibit their
          authetic work through a virtual gallery and social media integration.
          FusArt offers a distinctive service that combines artists&apos; work
          using AI to create original artworks while respecting artists&apos;
          intellectual property rights.
        </p>
        <p>
          Our goal is to provide a platform where artists can share their
          creativity with a global audience, encouraging them both financially
          and through acknowledgment.
        </p>
        <p>
          From traditional painting to digital art, we embrace all forms of
          artistic expression, fostering a supportive community to connect,
          collaborate, and support. By showcasing your work on FusArt, you gain
          exposure, connect with art enthusiasts, and have the opportunity to
          sell your art.
        </p>
      </section>
    </article>
  );
}
```

Copyright:


```
export default function About() {
  return (
    <article id="about" style={{ marginBottom: "12px"}}>
      <h2 className="is-title">about us</h2>
      <section className="divide-y-3">
        <p>
          FusArt is a fun and innovative way for artists to exhibit their
          authetic work through a virtual gallery and social media integration.
          FusArt offers a distinctive service that combines artists&apos; work
          using AI to create original artworks while respecting artists&apos;
          intellectual property rights.
        </p>
        <p>
          Our goal is to provide a platform where artists can share their
          creativity with a global audience, encouraging them both financially
          and through acknowledgment.
        </p>
        <p>
          From traditional painting to digital art, we embrace all forms of
          artistic expression, fostering a supportive community to connect,
          collaborate, and support. By showcasing your work on FusArt, you gain
          exposure, connect with art enthusiasts, and have the opportunity to
          sell your art.
        </p>
      </section>
    </article>
  );
}
```

```
            href="https://twitter.com/@user?s=20"
            target="_blank"
            rel="noreferrer"
            className="is-link"
          >
            <i className="fa-brands fa-x-twitter"> </i>
            X
          </a>

          <a
            href="mailto:fusart@gmail.com"
            className="is-link"
          >
            <i className="fa-solid fa-envelope"> </i>
            Email
          </a>
        </section>

        <section className="useful-links">
          <h2>Useful links</h2>

          <Link to="/acc/login" className="is-link">
            Login
          </Link>

          <Link to="/acc/register" className="is-link">
            Register
          </Link>

          <a href="#about" className="is-link">
            About Us
          </a>
        </section>
      </footer>
    );
  }
```

Home Entry:

- Flipping image
- Counter
- Input
- Button if Logged in or Not

```
export default function HomeEntry() {
  const { isLoggedIn } = useUserState();

  return (
    <div className="home-entry">
      <section className="img-flip">
        <img src="/img/art3.jpg" alt="" width="100%" />
      </section>

      <section className="welcomer">
        <div style={{ width: "fit-content" }}>
          <article className="">
            <span id="art-counter" className="main-title">1500</span>
             pieces of art as of today
          </article>

          <p style={{
            color: "var(--clr-prime)",
            fontWeight: "500",
            fontSize: "clamp(16px, 4vw, 28px)"
          }} >Let&apos;s draw up something for you</p>
          <input type="text" placeholder="get creative..." />

          <div className="center-it" style={{ maxWidth: "350px" }}>
            {isLoggedIn ? (
              <button onClick={() => console.log("Paintings fused!")} className="btn-fuse">FUSE</button>
            ) : (
              <Link to="/acc/login" className="btn-fuse">FUSE</Link>
            )}
          </div>
        </div>
      </section>
    </div>
  )
}
```

Login Structure:

```
export default function LoginStruct({
  children,
  title,
  fields = [],
  btnType = 'button',
  btnCallback = undefined,
  btnLabel,
  fallback = 'login',
  removeDefaultBtn = false,
}) {
  return (
    <div className="login-form step-1">
      <h2>{title}</h2>

      {fields.map((field, idx) => (
        <section key={idx}>
          <label htmlFor={field.id}>{field.label}</label>

          {/* simple state control */}
          <input
            id={field.id}
            type={field.type}
            required={field?.required ?? false}
            value={field.state}
            maxLength={field?.maxLength}
            minLength={field?.minLength}
            onChange={e => field.setState(e.target.value)}
          />
        </section>
      ))}
```

```jsx
{/* the children component property is preserved if you want to add additional custom JSX inside */}
{children}

{!removeDefaultBtn && (
  <section className="submit-sect">
    <button type={btnType} onClick={btnCallback}>
      {btnLabel}
    </button>
  </section>
)}

<p className="fallback-msg">
  {fallback === 'register' ? (
    <>
      Already have an account? 
      <Link to="/acc/login" className="is-link">
        login
      </Link>
    </>
  ) : (
    <>
      Don&apos;t have an account? 
      <Link to="/acc/register" className="is-link">
        register
      </Link>
    </>
  )}
</p>
</div>
```

Navigation bar:

- Top Bar
- Link to login or Register
- Profile

Before login

```jsx
export default function Navbar() {
  const { isLoggedIn, logoutUser } = useUserState();

  return (
    <nav className="navbar">
      <section>
        <Link to="/" className="nav-item">
          <img src="/img/logo.jpg" alt="" width={30} height={30} />
        </Link>
      </section>

      <section style={{ display: "flex" }}>
        {/* conditionally render buttons and links */}
        {!isLoggedIn ? (
          <>
            <input type="checkbox" id="dropdown-toggler" style={{ display: "none" }} />

            <label htmlFor="dropdown-toggler" className="nav-item">
              <i className="fa-solid fa-bars"></i>
              <i className="fa-solid fa-xmark"></i>
            </label>

            <div className="dropdown">
              <Link to="/acc/login" className="nav-item">LOGIN</Link>
              <Link to="/acc/register" className="nav-item">REGISTER</Link>
            </div>

            <Link to="/acc/login" className="nav-item acc">LOGIN</Link>
            <Link to="/acc/register" className="nav-item acc">REGISTER</Link>
          </>
        ) : (
```

After login:

```
            <label htmlFor="dropdown-toggler" className="nav-item">
              My Account 
              <i className="fa-solid fa-chevron-down"></i>
            </label>

            <div className="dropdown">
              <Link to="/acc/profile" className="nav-item">
                <i className="fa-solid fa-user"></i> 
                PROFILE
              </Link>
              <button onClick={logoutUser} className="nav-item">
                <i className="fa-solid fa-right-from-bracket"></i> 
                LOGOUT
              </button>
            </div>
          </>
        )}
      </section>
    </nav>
  )
}
```

Photo slider:

```
export default function PhotoSlider() {
  /**
   * note: if you want to add extra images, enclose each 1 between a <SplideSlide> tag
   * like the pattern below.
   * the image srcs should be inside the public/img folder.
   */
  return (
    <div id="photo-slider">
      <h2 className='is-title'>Explore world-class art</h2>

      <Splide
        aria-label="My Favorite Images"
        options={{
          rewind: true,
          perMove: 1,
          gap: '12px',
          perPage: 3,

          // the breakpoints works like (max-width: 768px) desktop-first
          breakpoints: {
            768: {
              perPage: 1,
            },
            1024: {
              perPage: 2,
            },
          },
        }}
      >
        <SplideSlide>
          <img src="/img/art3.jpg" alt="" />
        </SplideSlide>

        <SplideSlide>
          <img src="/img/art2.jpg" alt="" />
        </SplideSlide>

        <SplideSlide>
          <img src="/img/art1.jpg" alt="" />
        </SplideSlide>
      </Splide>
    </div>
  );
}
```

Register step 2:

```
export default function RegisterStepTwo({ role, setRole, bio, setBio, goBack, goNext }) {
  return (
    // I made the LoginStruct flexible so I can manipulate it freely
    <LoginStruct
      title="Your art journey"
      removeDefaultBtn
      fallback="register"
    >
      {/* custom (non-supported) fields */}
      <div>
        <label>Role </label>

        <section className="multiple-choice-field">
          <div>
            <input
              id="role-user"
              type="radio"
              value="user"
              checked={role === "user"}
              onChange={e => setRole(e.target.value)}
            />
            <label htmlFor="role-user">user</label>
          </div>

          <div>
            <input
              id="role-artist"
              type="radio"
              checked={role === "artist"}
              value="artist"
              onChange={e => setRole(e.target.value)}
            />
            <label htmlFor="role-artist">artist</label>
          </div>
        </section>
      </div>
```

```
        <section>
          <label htmlFor="bio">Bio</label>
          <textarea
            id="bio"
            value={bio}
            onChange={e => setBio(e.target.value)}
            style={{ resize: "vertical" }}
          ></textarea>
        </section>


        {/* buttons */}
        <section className="submit-sect">
          <button type="button" onClick={goBack}>Go back</button>
          <button type="button" onClick={goNext}>Go to step 3</button>
        </section>
      </LoginStruct>
    )
  }
```

Register step 3:

```
export default function RegisterStepThree({
  setArtTypes,
  setArtStyles,
  goBack,
}) {
  const handleCheckbox = (e, setState) => {
    // `other` is the current array
    if (e.target.checked) {
      setState(other => [...other, e.target.value]);
    } else {
      // the filter function will remove the unchecked art type
      setState(other => other.filter(type => type !== e.target.value));
    }
  };

  // these 2 arrays makes work more efficient
  const ART_TYPES = ['traditional', 'digital'];
  const ART_STYLES = [
    'abstract',
    'minimalism',
    'surrealism',
    'impressionism',
    'realism',
    'cubism',
    'pop_art',
  ];

  return (
    <LoginStruct
      title="Your interests"
      removeDefaultBtn
      fallback="register"
      withPolicyAgreement
    >
```

```
    <div>
      <label>Types of visual art </label>

      <section className="multiple-choice-field">
        {ART_TYPES.map((type, idx) => (
          <div key={idx}>
            <input
              id={`art-type-${type}`}
              type="checkbox"
              value={type}
              onChange={e => handleCheckbox(e, setArtTypes)}
            />

            <label htmlFor={`art-type-${type}`}>{type}</label>
          </div>
        ))}
      </section>
    </div>

    <div>
      <label>Styles of visual art </label>

      <section className="multiple-choice-field">
        {ART_STYLES.map((type, idx) => (
          <div key={idx}>
            <input
              id={`art-style-${type}`}
              type="checkbox"
              value={type}
              onChange={e => handleCheckbox(e, setArtStyles)}
            />

            <label htmlFor={`art-style-${type}`}>{type}</label>
          </div>
        ))}
      </section>
    </div>
```

```
    {/* policy agreement */}
    <section style={{ marginTop: '12px' }}>
      <input
        id="policy-agree"
        type="checkbox"
        required
      />
      <label htmlFor="policy-agree" style={{ display: 'inline' }}>
        <b>Agree to terms and conditions and privacy policy</b>
      </label>
    </section>

    {/* buttons */}
    <section className="submit-sect">
      <button type="button" onClick={goBack}>
        Go back
      </button>
      <button type="submit">REGISTER</button>
    </section>
  </LoginStruct>
  );
}
```

Features

Hero section

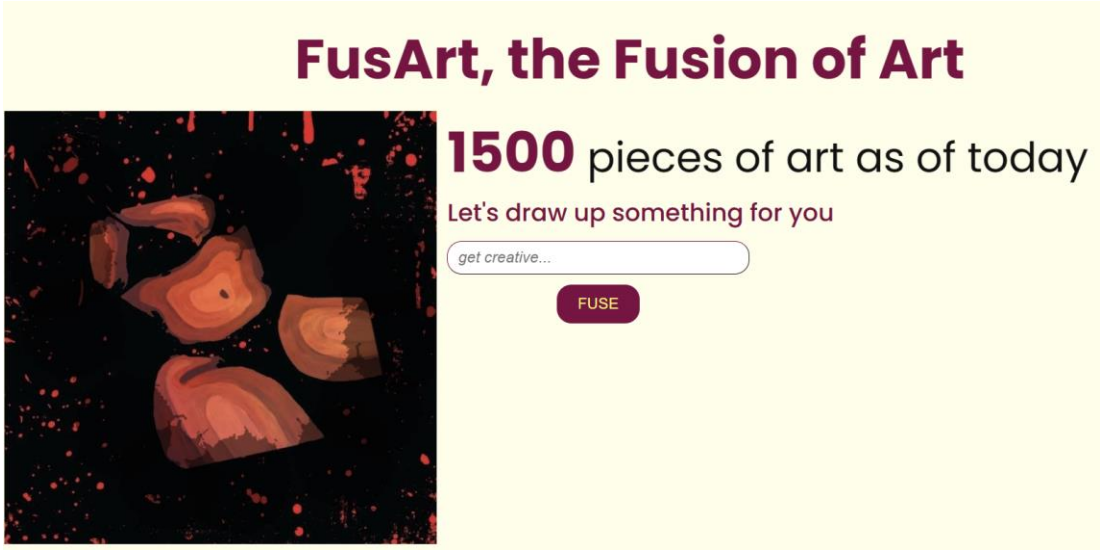
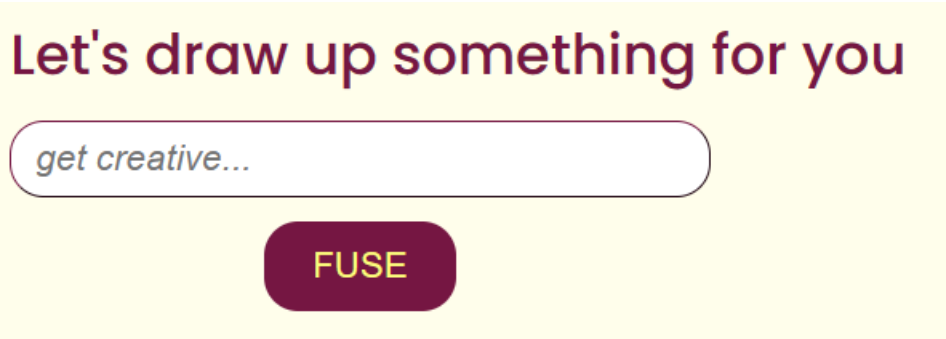
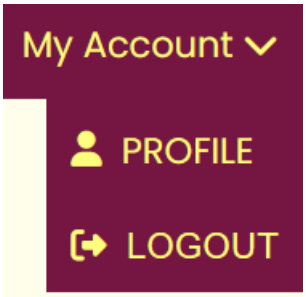


Photo slider

Search engine:



Drop down menu

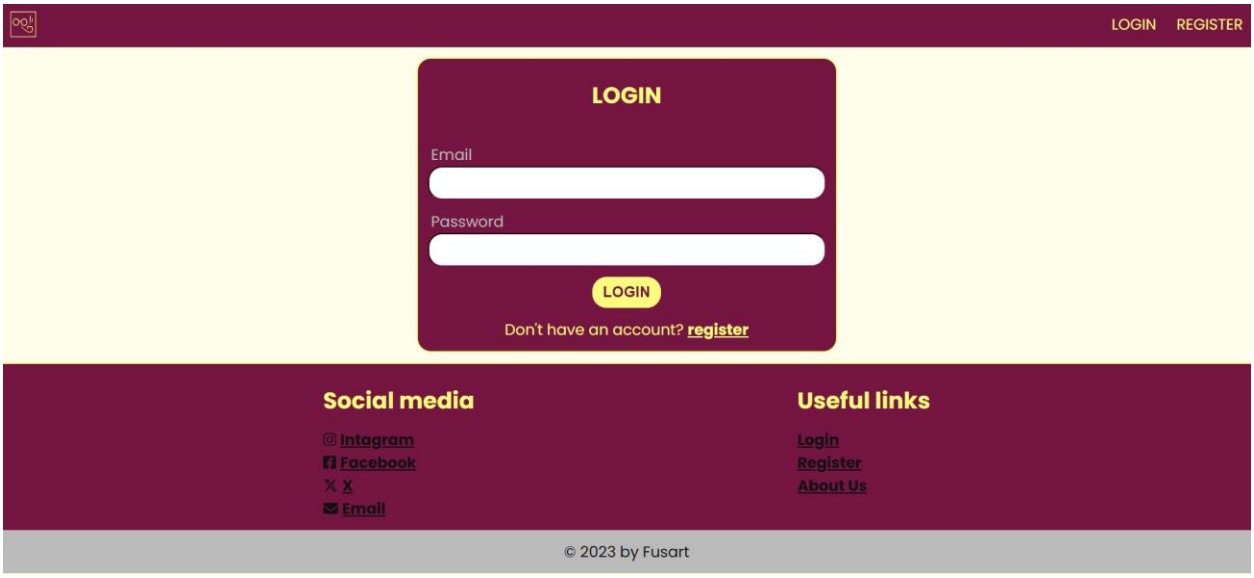


Footer and copyright

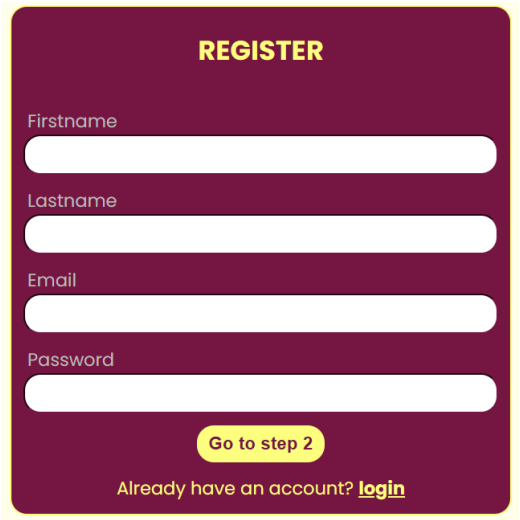
Cards: account info

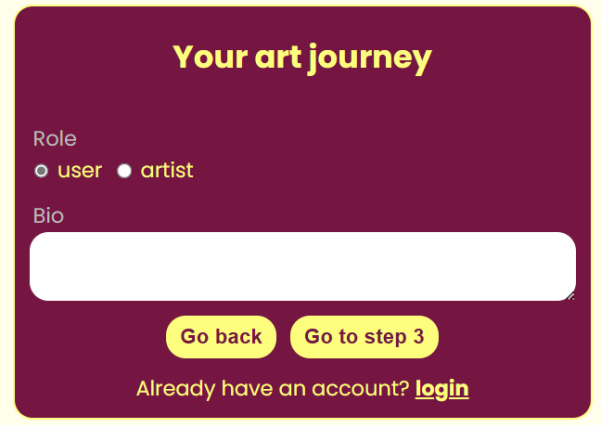
Firstname	oriana
Lastname	eid
Email	orianaaeid@gmail.com
Role	artist
Bio	specialized in minimalist art
Art types	traditional
Art styles	minimalism, pop_art
Registered	24/11/2023

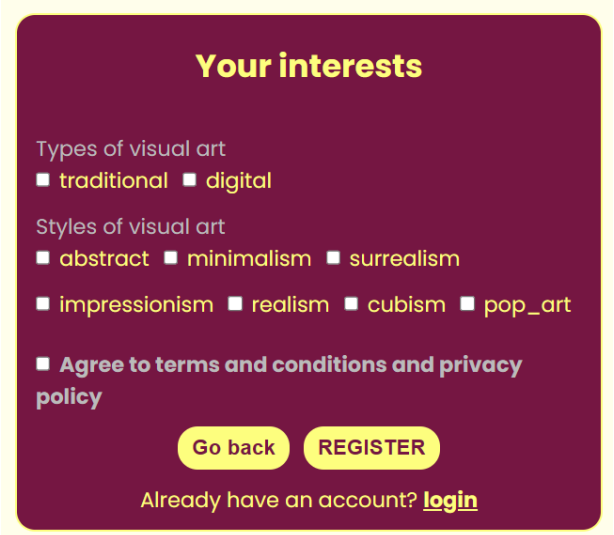
User login:



User sign up (with frontend verification)







Conclusion

Future improvements:

- Add some components: search (need for database)
- Implement the main feature: the AI model
- Augment the database