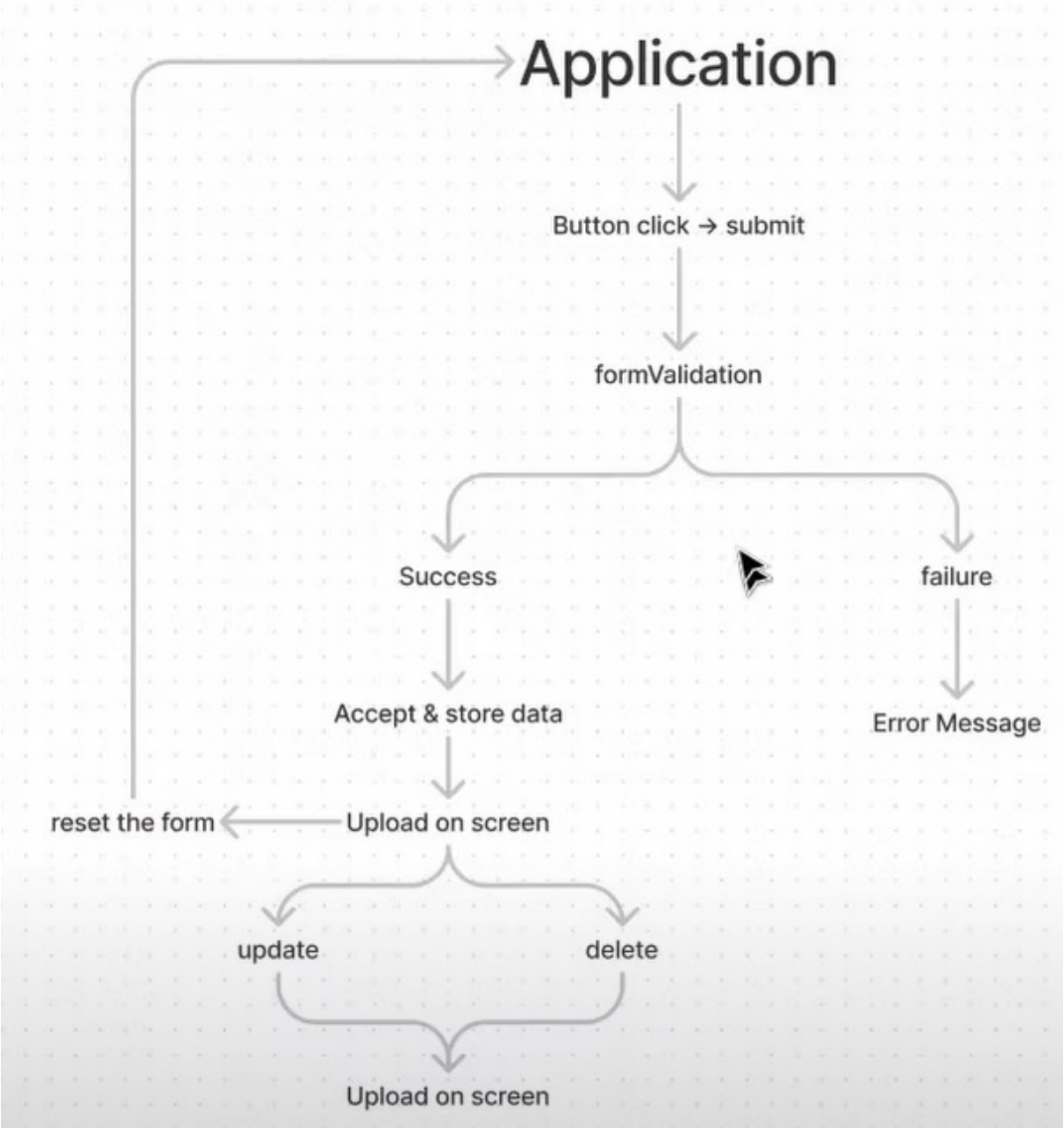
Essay outline

TO DO APLICATION

The crowd application allows the user to create, read and update data.
That data at the same time gives the power so the users to delete that data as well.



The very first thing that we have is called the **Application** which is actually our social media app.

Then we have something called **the button click submit.** So what this will do is whenever the user clicks this post is gonna go to go and trigger a function called **form validation** and this function will have two ends either it's gonna be successful or is gonna be a failure. So why success or why failure? That's because when our user submits the form blank without writing anything here then we're gonna go to failure state. At the same time we're gonna get something called an error message. The error message will come here in red color. It's gonna say your message cannot be blank. If our user writes something here and if they post it then we're gonna go to the success state. That success state is gonna trigger all of these functions that are coming after the success state.

So now we are target the form here with the id 'form' . Let's target this inside javaScript. So in main.js do this

```
let form = document.getElementById("form");
```

If we look at the PICTURE first of all we have the button click so let's make that.

First of all target the form and add an even listener something like this

form.addEventListener("submit")

So which event listener am i using it's called the submit . Why that is because this is our submit button.

Then i'm gonna make callback function, it's gonna be an arrow function

Now to check whether this button click event is working or not let's

console.log('button clicked')

Now i'm going to click the post but look at this behavior, every time i post here I see this it's refreshing automatically, we need to prevent this so we're going to use something called prevent default

```
form.addEventListener("submit", (e) => {
  e.preventDefault();
  console.log("button clicked");
```

Whenever we click this button submit it should trigger a function called formValidation.

So let's build that function. I'm going to make a function validation

```
let formValidation = () => {
  if (input.value === "") {
    msg.innerHTML = "Post cannot be blank";
    console.log("failure");
```

```
  } else {
    console.log("success");
```

How many states do i have 2=> success and the failure state which means that i'm gonna use what an if else statement. So on which basis are we building this up?
On the basis of whether our user is actually submitting this blank or are they writing something and then submitting that. So in order to target this text area, what i'm gonna do is I'm gonna come back to my html and find the text area which id is "input" so i'm going to copy that and go back to the main.js now I am gonna target the input like this

```
let input = document.getElementById("input");
```

Now if our input is blank then i'm gonna write here

```
console.log("failure");
```

but if the form is not blank then I'm gonna come to this else and write here success

```
 console.log("success");
```

No I'm gonna write the input.value here triple to sign then make it blank, which will means that if our input is blank then show this one that we are on a failure state but if it is not blank then show success.
It looks like this

```
let formValidation = () => {
  if (input.value === "") {
      console.log("failure");
  } else {
    console.log("success");
  }
};
```

Now we should invoke the function

```
 formValidation();
```

We should invoke it inside here, so every time we click on button post it will run

```
form.addEventListener("submit", (e) => {
  e.preventDefault();
  console.log("button clicked");
  formValidation();
});
```

Now i'm gonna come here on my results screen and I'm submitting it, so how will

the users know whether this is the success state or the failure state. For that reason we're gonna give an error message. I'm going to come back to my index.html and below text area make a div with id="msg" . In style.css I'm gonna make it in the color of red it means the error color. Now go back to main.js and do this

```
let formValidation = () => {
  if (input.value === "") {
    msg.innerHTML = "Post cannot be blank";//this is that message
    console.log("failure");
  } else {
    console.log("success");
    msg.innerHTML = "";//when whrite something, RED colored message to be removed
    acceptData();
  }
};
```

Now what we're gonna do is we're gonna **accept and store** the data.
First we're gonna make a empty object let data = {}

```
let data = {};
```

Now let's make a function

```
let acceptData = () => { //now to push data inside this object=>
  data["text"] = input.value;// to collect the text from text area
  console.log(data);
  createPost();
};
```

To understand what's actually happening using this function and using this empty object so i'm going to come back to my results screen so look at this=>whatever i write in the text area I'm going to collect that using this function an then we're going to push it inside this data
Now we need to **invoke this function acceptData()** , We are invoke it inside success state **inside function formValidation**
To see if this works open e console and now I'm gonna post it blank look every time we click the button post in console shows=> button clicked=>failure (because its blank post)
Now let's write it something in textarea, and if we open the console shows=>success=>data pushed.
Now we're going to upload the data in the screen. We will use funcion for that.
First we are target the posts here with the id ='posts' . Let's target this inside javaScript.
So in main.js do this

```
let posts = document.getElementById("posts");
```

```
let createPost = () => { //we should invoke createPost() inside function acceptData
  posts.innerHTML += `// WE put a plus += here so the previous post not get erased
    <div>
        <p>${data.text}</p>
        <span class="options">
            <i onClick = "editPost(this)" class="fa-solid fa-pen-to-square"></i>
            <i onClick = "deletePost(this)" class="fa-solid fa-trash-can"></i>
        </span>
    </div>
    `;
  input.value = "";
};
```

Now when we write something in the textArea, is updateing as a post on the screen.
Now we need to build two more functions **The update** and **The delete**
Now i'm gonna write here a function **deletePosts**

```
let deletePost = (e) => {
  e.parentElement.parentElement.remove();
  input.value = "";
};
```

We use **this** to target only one piece wherever we put our mouse. Let's say I have put my mouse here on the first one not on the second one if i click on this it's gonna delete only this one not the others.
Now if I write something in the textArea and click delete button, its gonna delete only the icon, not the whole text. That's because we are only targeting only onesingle icon so we need to delete this entire parent . If I write e.parentElement.remove() It willl be deleted only the icon sets, it is not deleting the entire thing which means that we are only targeting the span not the main parent. Thats why we are targeting the parent div too, and should write e.parentElement.parentElement.remove();Now it's gonna delete the entire thing .
Now I'm gonna make a new function for editing post.

```
let editPost = (e) => {
  input.value = e.parentElement.previousElementSibling.innerHTML;
  e.parentElement.parentElement.remove();
};
```