PROJECT NAME: SIMPLYBLOGGING SITE

Tools and Technology Used:

1. HTML
2. CSS
3. Javascript
4. Bootstrap
5. Visual Studio Code
6. GitHub/Git

HTML files:

1.index.html

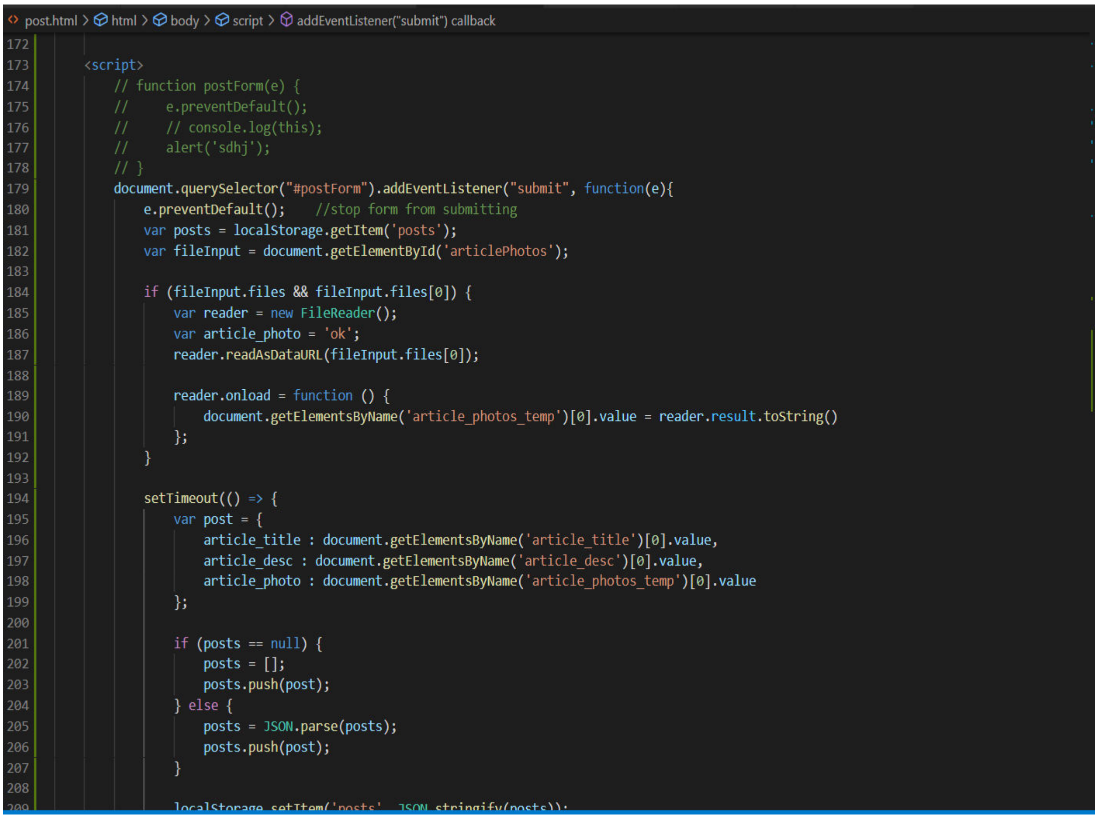
2.about.html

3.login.html

4.post.html

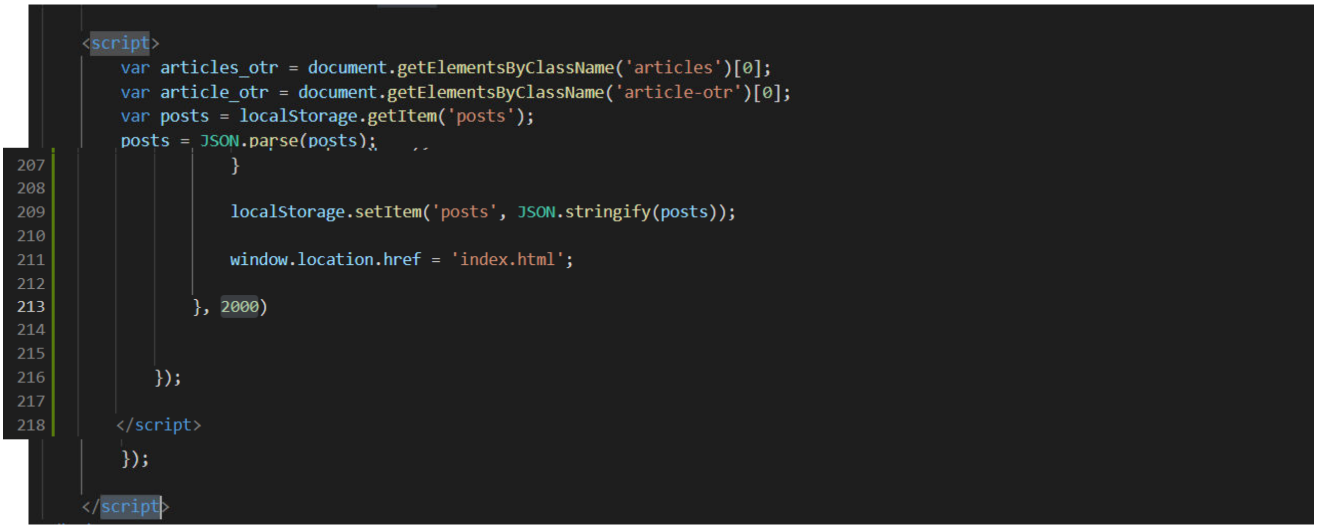
Internal Javascript has been used for the functionality of adding new blog posts:

screenshots:



post.html > html > body > script > addEventListener("submit") callback

```
    <script>
        // function postForm(e) {
        //     e.preventDefault();
        //     // console.log(this);
        //     alert('sdhj');
        // }
        document.querySelector("#postForm").addEventListener("submit", function(e){
            e.preventDefault();    //stop form from submitting
            var posts = localStorage.getItem('posts');
            var fileInput = document.getElementById('articlePhotos');

            if (fileInput.files && fileInput.files[0]) {
                var reader = new FileReader();
                var article_photo = 'ok';
                reader.readAsDataURL(fileInput.files[0]);

                reader.onload = function () {
                    document.getElementsByName('article_photos_temp')[0].value = reader.result.toString()
                };
            }

            setTimeout(() => {
                var post = {
                    article_title : document.getElementsByName('article_title')[0].value,
                    article_desc : document.getElementsByName('article_desc')[0].value,
                    article_photo : document.getElementsByName('article_photos_temp')[0].value
                };

                if (posts == null) {
                    posts = [];
                    posts.push(post);
                } else {
                    posts = JSON.parse(posts);
                    posts.push(post);
                }

                localStorage.setItem('posts', JSON.stringify(posts));
```



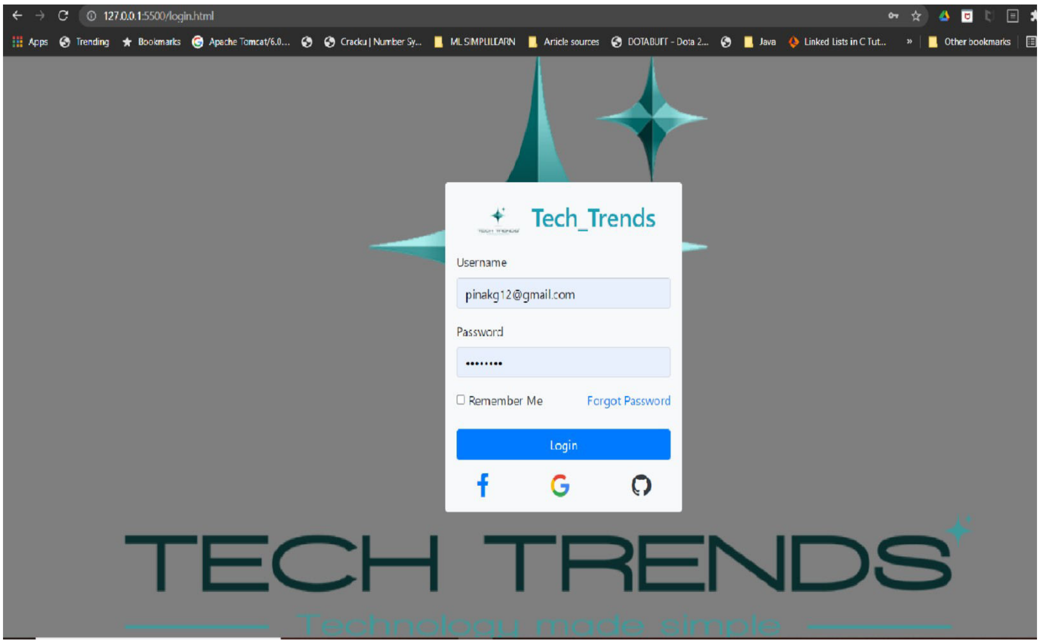
```
<script>
    var articles_otr = document.getElementsByClassName('articles')[0];
    var article_otr = document.getElementsByClassName('article-otr')[0];
    var posts = localStorage.getItem('posts');
    posts = JSON.parse(posts);
                }

                localStorage.setItem('posts', JSON.stringify(posts));

                window.location.href = 'index.html';

            }, 2000)


        });

    </script>
    });
</script>
```

External JS used for login functionality:

File Name: login.js

Process:

1.Login page has been created using the HTML file : login.html
HTML `<form>` element is used to create an HTML form for the login page.
Screenshot:



127.0.0.1:5500/login.html

Tech_Trends

Username: pinakg12@gmail.com

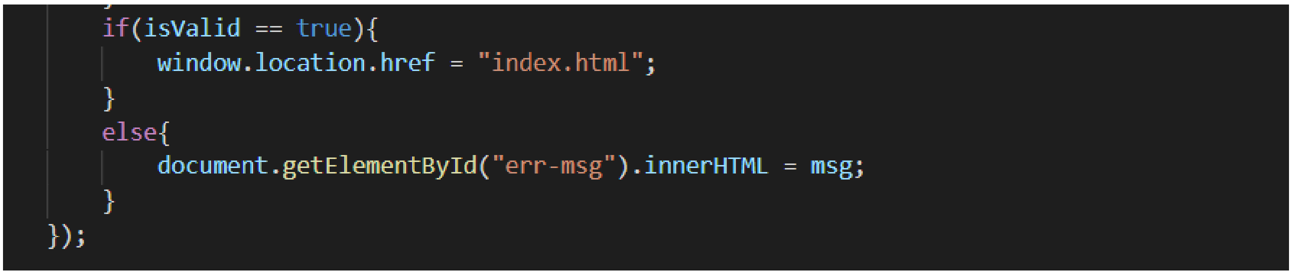
Password: ••••••••

□ Remember Me   Forgot Password

Login

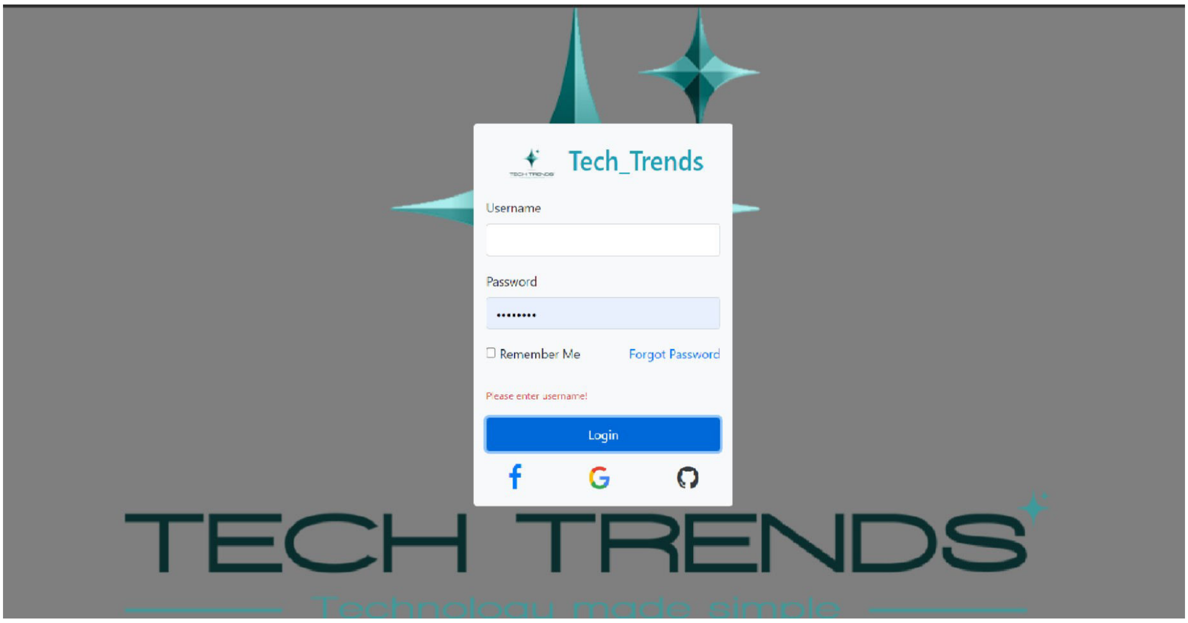
When the user clicks on the login button , the form is sumbitted using addEventListener() method which is fired when the user clicks the login button .
There are validations for every form element which has been handled in login.js .
When the username is left blank , error will be displayed as "Please enter username! "
When the Password is left blank , error will be displayed as "Please enter password! "
When both the fields are left blank , error will be displayed as "Please enter username and paswoord! "
The Password should contain atleast 1 uppercase, 1 lowercase, 1 digit and 1 special character!
Otherwise error will be displayed as "Please make sure that your password contain atleast 1 uppercase, 1 lowercase, 1 digit and 1 special character! "

When all the form fields are validated user will be redirected to the Home page (index.html) using window.location.href.



```
    if(isValid == true){
        window.location.href = "index.html";
    }
    else{
        document.getElementById("err-msg").innerHTML = msg;
    }
});
```

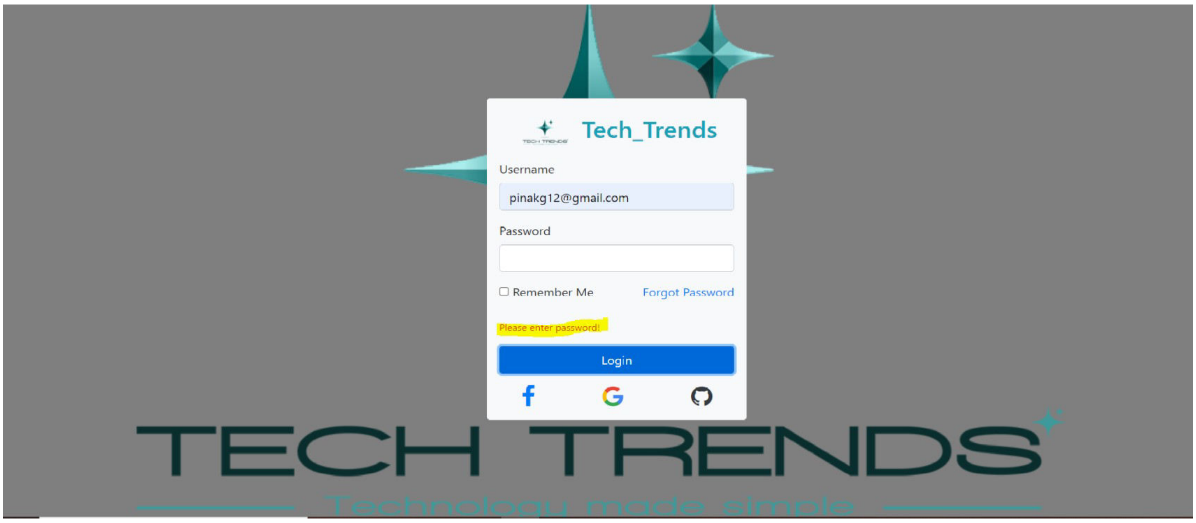
ERROR SCREENSHOTS:



Username

Password
••••••••

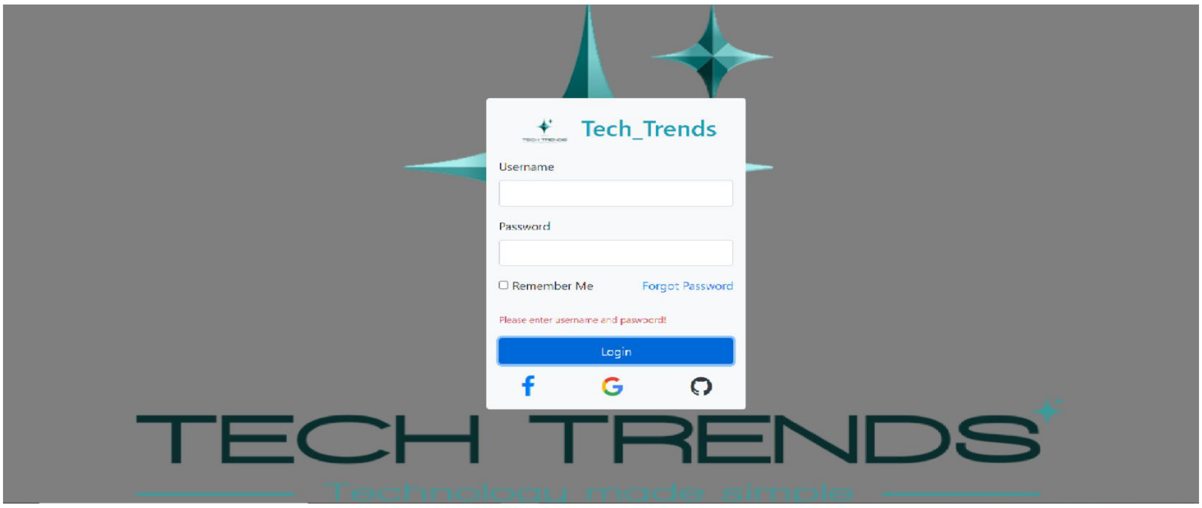
Please enter username!



Username
pinakg12@gmail.com

Password

Please enter password!



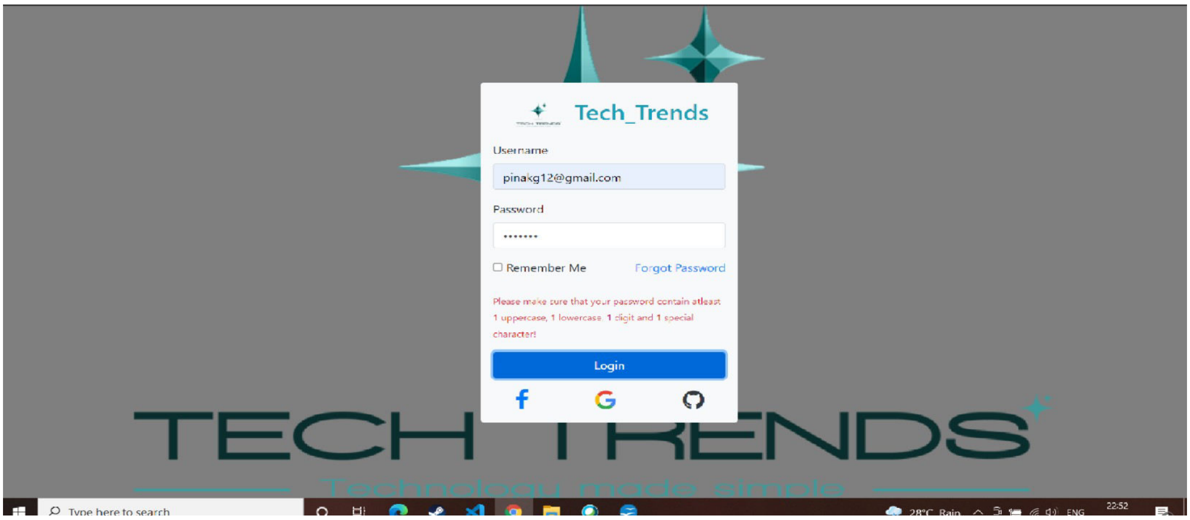
Username

Password

□ Remember Me Forgot Password

Please enter username and password!

Login



Username: pinakg12@gmail.com

Password: •••••••

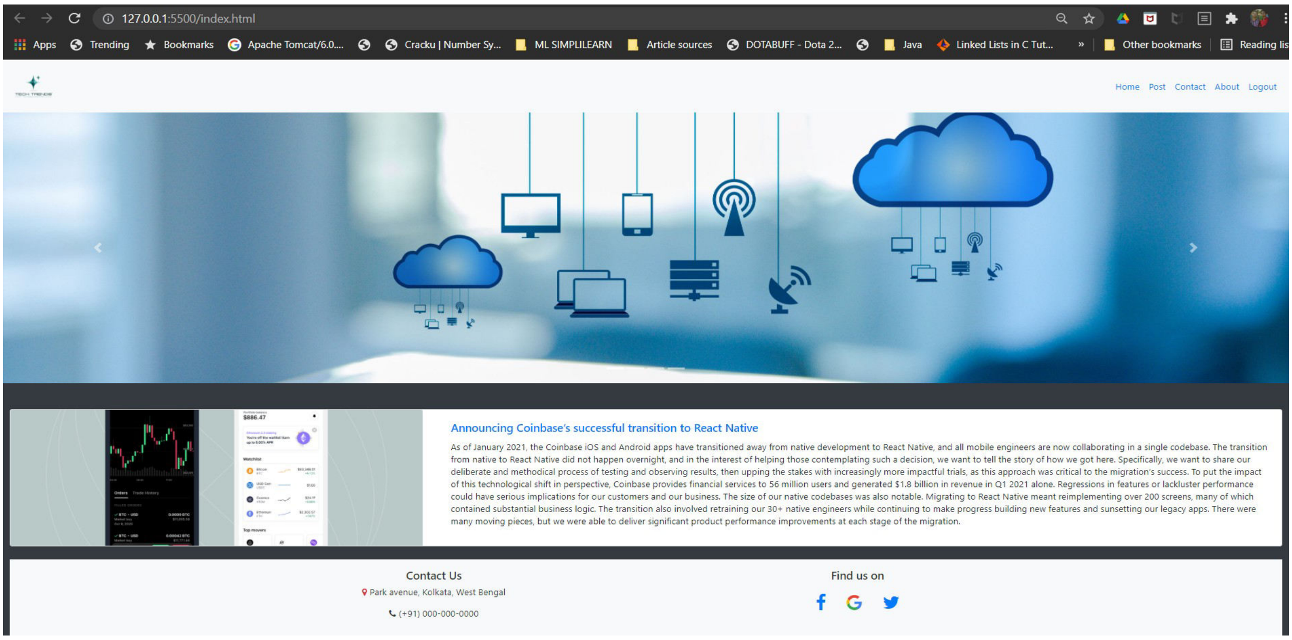
□ Remember Me Forgot Password

Please make sure that your password contain atleast 1 uppercase, 1 lowercase, 1 digit and 1 special character!

Login

2.index.html is created for the Home page containing all the blogs.

SCREENSHOT:



Announcing Coinbase's successful transition to React Native

As of January 2021, the Coinbase iOS and Android apps have transitioned away from native development to React Native, and all mobile engineers are now collaborating in a single codebase. The transition from native to React Native did not happen overnight, and in the interest of helping those contemplating such a decision, we want to tell the story of how we got here. Specifically, we want to share our deliberate and methodical process of testing and observing results, then upping the stakes with increasingly more impactful trials, as this approach was critical to the migration's success. To put the impact of this technological shift in perspective, Coinbase provides financial services to 56 million users and generated $1.8 billion in revenue in Q1 2021 alone. Regressions in features or lackluster performance could have serious implications for our customers and our business. The size of our native codebases was also notable. Migrating to React Native meant reimplementing over 200 screens, many of which contained substantial business logic. The transition also involved retraining our 30+ native engineers while continuing to make progress building new features and sunsetting our legacy apps. There were many moving pieces, but we were able to deliver significant product performance improvements at each stage of the migration.

Contact Us

Park avenue, Kolkata, West Bengal

(+91) 000-000-0000

Find us on

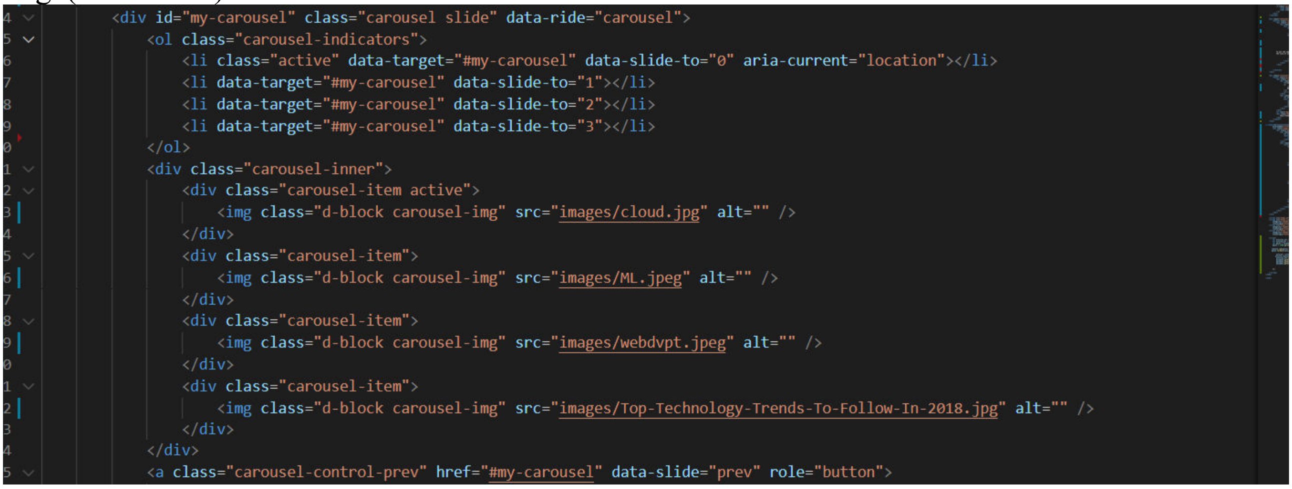
Navbar:

A grey horizontal navbar that becomes vertical on small screens

```
<nav class="navbar navbar-expand-sm bg-light">
```

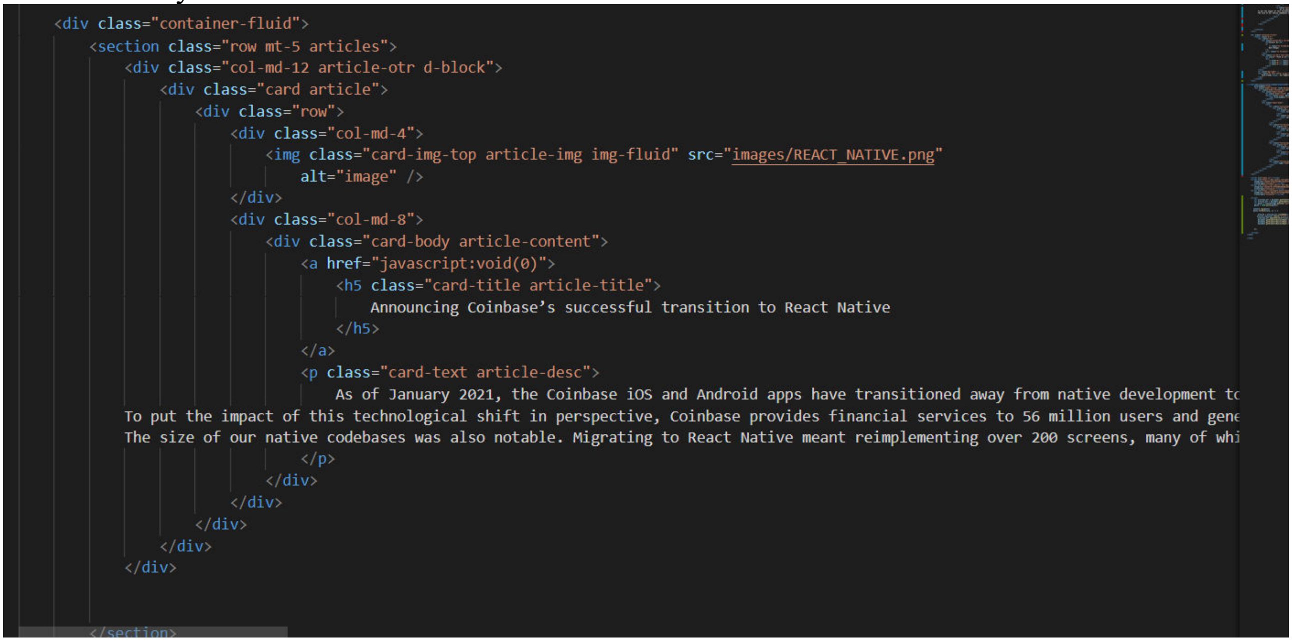
Link has been created inside unordered lists for Home,Post,Contact,About,Logout using the <a> tag (anchor tag).

Bootstrap carousel plugin has been used to slide through a series of images on the Home Page(index.html)



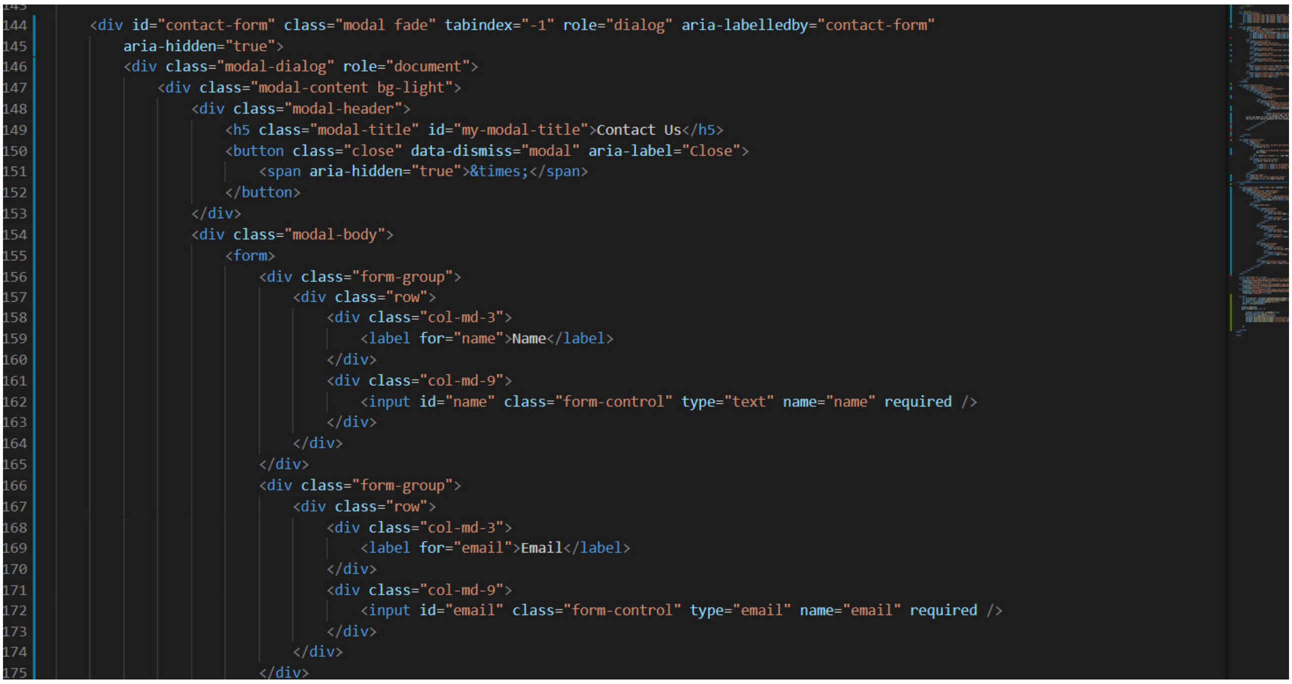
```
<div id="my-carousel" class="carousel slide" data-ride="carousel">
    <ol class="carousel-indicators">
        <li class="active" data-target="#my-carousel" data-slide-to="0" aria-current="location"></li>
        <li data-target="#my-carousel" data-slide-to="1"></li>
        <li data-target="#my-carousel" data-slide-to="2"></li>
        <li data-target="#my-carousel" data-slide-to="3"></li>
    </ol>
    <div class="carousel-inner">
        <div class="carousel-item active">
            <img class="d-block carousel-img" src="images/cloud.jpg" alt="" />
        </div>
        <div class="carousel-item">
            <img class="d-block carousel-img" src="images/ML.jpeg" alt="" />
        </div>
        <div class="carousel-item">
            <img class="d-block carousel-img" src="images/webdvpt.jpeg" alt="" />
        </div>
        <div class="carousel-item">
            <img class="d-block carousel-img" src="images/Top-Technology-Trends-To-Follow-In-2018.jpg" alt="" />
        </div>
    </div>
    <a class="carousel-control-prev" href="#my-carousel" data-slide="prev" role="button">
```

Bootstrap **container-fluid** element has been used in the div class for the blog posts to   resize accordingly as you make even the smallest changes to your browser width. This makes the page reactive to any browser window size.



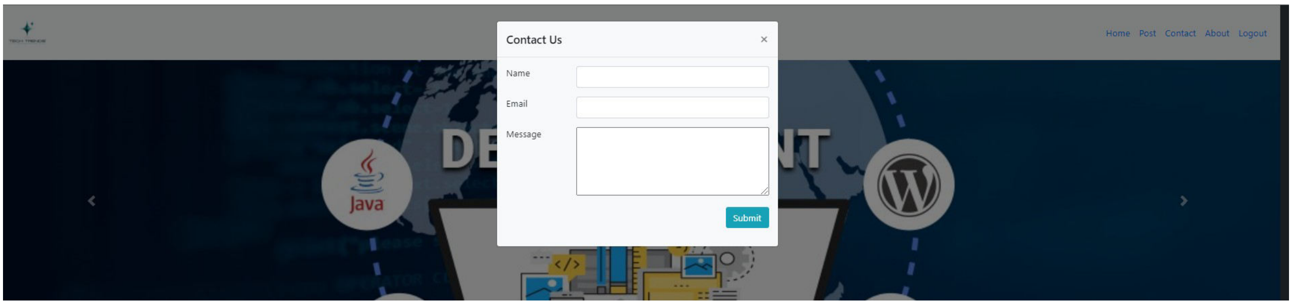
```html
<div class="container-fluid">
    <section class="row mt-5 articles">
        <div class="col-md-12 article-otr d-block">
            <div class="card article">
                <div class="row">
                    <div class="col-md-4">
                        <img class="card-img-top article-img img-fluid" src="images/REACT_NATIVE.png"
                            alt="image" />
                    </div>
                    <div class="col-md-8">
                        <div class="card-body article-content">
                            <a href="javascript:void(0)">
                                <h5 class="card-title article-title">
                                    Announcing Coinbase’s successful transition to React Native
                                </h5>
                            </a>
                            <p class="card-text article-desc">
                                As of January 2021, the Coinbase iOS and Android apps have transitioned away from native development tc
        To put the impact of this technological shift in perspective, Coinbase provides financial services to 56 million users and gene
        The size of our native codebases was also notable. Migrating to React Native meant reimplementing over 200 screens, many of whi
                            </p>
                        </div>
                    </div>
                </div>
            </div>
        </div>


    </section>
```

Modal has been used for the Contact Form when user clicks on Contact.

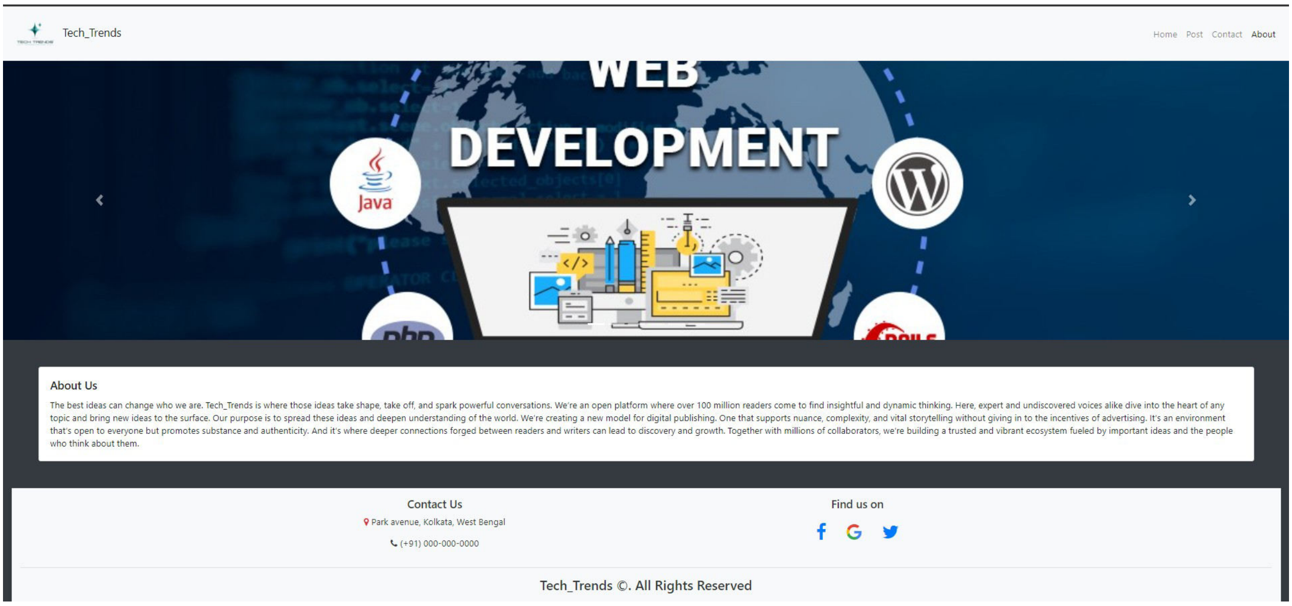


```html
<div id="contact-form" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="contact-form"
    aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content bg-light">
            <div class="modal-header">
                <h5 class="modal-title" id="my-modal-title">Contact Us</h5>
                <button class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <form>
                    <div class="form-group">
                        <div class="row">
                            <div class="col-md-3">
                                <label for="name">Name</label>
                            </div>
                            <div class="col-md-9">
                                <input id="name" class="form-control" type="text" name="name" required />
                            </div>
                        </div>
                    </div>
                    <div class="form-group">
                        <div class="row">
                            <div class="col-md-3">
                                <label for="email">Email</label>
                            </div>
                            <div class="col-md-9">
                                <input id="email" class="form-control" type="email" name="email" required />
                            </div>
                        </div>
                    </div>
```

Contact Form :



3.About Page has been created using HTML file about.html, onclick About from Home user is redirected to the About page containing information about the Organization.



About Us

The best ideas can change who we are. Tech_Trends is where those ideas take shape, take off, and spark powerful conversations. We're an open platform where over 100 million readers come to find insightful and dynamic thinking. Here, expert and undiscovered voices alike dive into the heart of any topic and bring new ideas to the surface. Our purpose is to spread these ideas and deepen understanding of the world. We're creating a new model for digital publishing. One that supports nuance, complexity, and vital storytelling without giving in to the incentives of advertising. It's an environment that's open to everyone but promotes substance and authenticity. And it's where deeper connections forged between readers and writers can lead to discovery and growth. Together with millions of collaborators, we're building a trusted and vibrant ecosystem fueled by important ideas and the people who think about them.

Contact Us

Park avenue, Kolkata, West Bengal

☎ (+91) 000-000-0000

Find us on

Tech_Trends ©. All Rights Reserved

4.Adding Blogs by the User.

Logic Description:

AddEventListener() method has been used to add an event handler to the submit element.

FileReader.readAsDataURL() method has been used to read the contents of the File which contains the Image which is to be uploaded for the Blog Post.

When the read operation is finished, the readyState becomes DONE, and the loadend is triggered. At that time, the result attribute contains the data as a data: URL representing the file's data as a base64 encoded string.

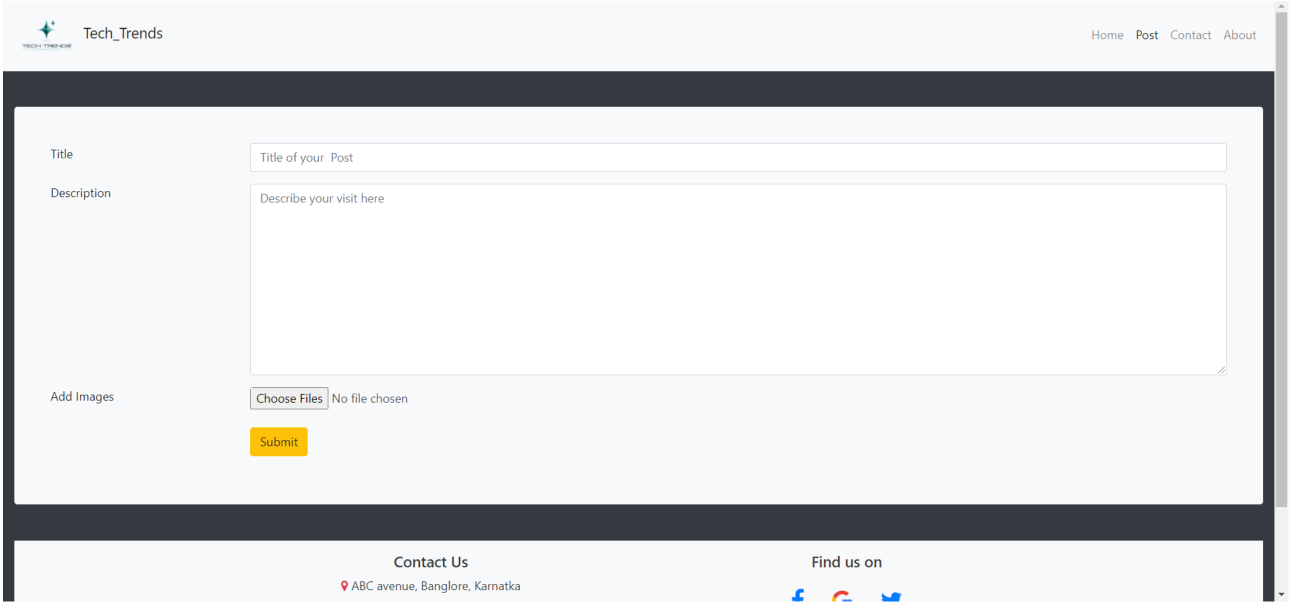
The localStorage object stores data for the new Blog Post which is to be added . The data will not be deleted when the browser is closed.

After storing the new blog data in localStorage , it is redirected to index.html .

When the user clicks on Post in Navbar user is redirected to post.html which contains the form containing input fields :

- Title
- Description
- Add Images

Title: User needs to add the title of the blog
Description: User need to add the description of the blog
Add Images: User need to add image for the blog post.

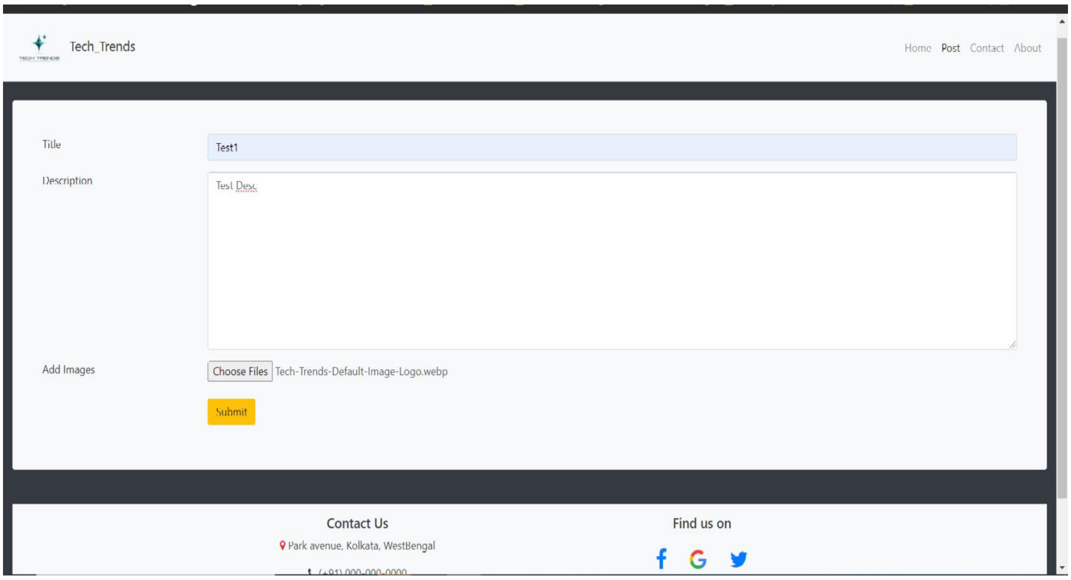


Title: Title of your Post

Description: Describe your visit here

Add Images: Choose Files No file chosen

After the user adds the input in the fields mentioned and clicks on submit, user is redirected to the Home page and the user blog gets added in the blogs.

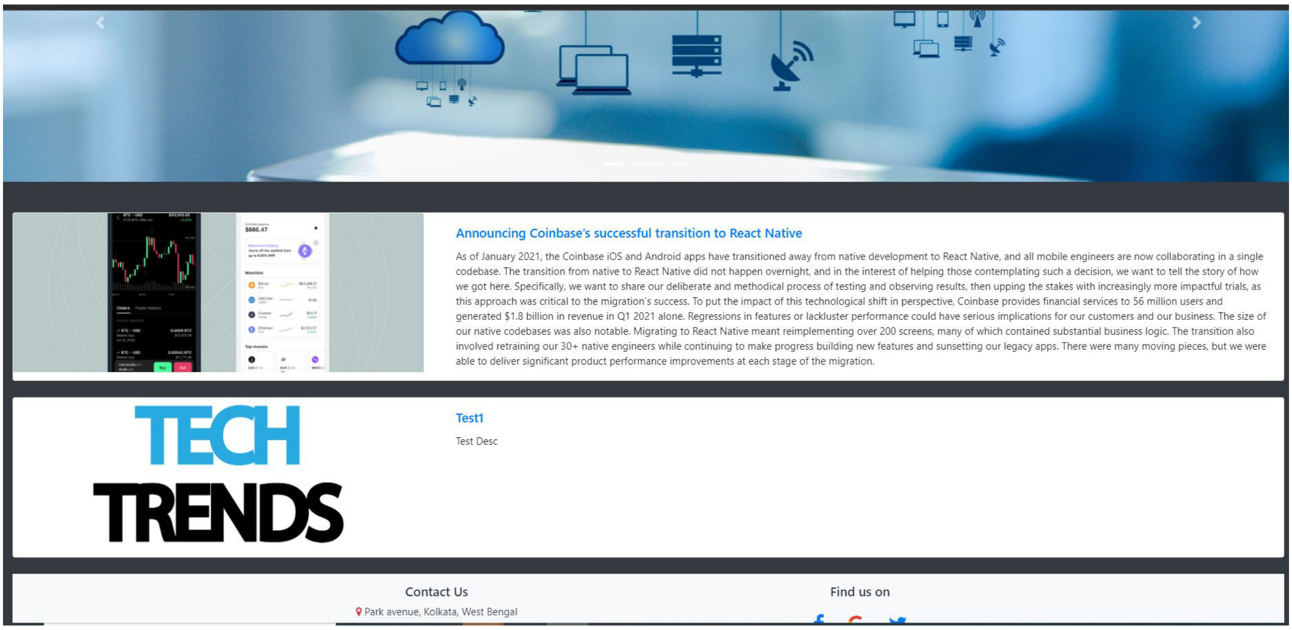


Title: Test1

Description: Test Desc

Add Images: Choose Files Tech-Trends-Default-Image-Logo.webp

Blog Post has now been added to the Home page.



Announcing Coinbase's successful transition to React Native

As of January 2021, the Coinbase iOS and Android apps have transitioned away from native development to React Native, and all mobile engineers are now collaborating in a single codebase. The transition from native to React Native did not happen overnight, and in the interest of helping those contemplating such a decision, we want to tell the story of how we got here. Specifically, we want to share our deliberate and methodical process of testing and observing results, then upping the stakes with increasingly more impactful trials, as this approach was critical to the migration's success. To put the impact of this technological shift in perspective, Coinbase provides financial services to 56 million users and generated $1.8 billion in revenue in Q1 2021 alone. Regressions in features or lackluster performance could have serious implications for our customers and our business. The size of our native codebases was also notable. Migrating to React Native meant reimplementing over 200 screens, many of which contained substantial business logic. The transition also involved retraining our 30+ native engineers while continuing to make progress building new features and sunsetting our legacy apps. There were many moving pieces, but we were able to deliver significant product performance improvements at each stage of the migration.

Test1

Test Desc

Contact Us

Park avenue, Kolkata, West Bengal

Find us on