Add 2FA support for login

GSOC ‘24

Personal Details	Description
Name	Manasvi Gaur
Organisation Details	College: Bennett University Program: Computer Science and Engineering Current Semester:4 Expected Graduation Date:2026
Contact Details	Phone No: 9511319440 Email ID: manasvigaur3@gmail.com
Current Timezone	UTC+5:30
Github Profile	Github
Linkedin Profile	LinkedIn
Resume	Resume

Project Overview

Two-factor authentication (2FA) adds an extra layer of security to the authentication process by making it harder for attackers to access a person's devices or online accounts. 2FA provides more protection against unauthorised access to user accounts and reduces fraud risks.

Our project aims to significantly enhance the security of our platform's login process by integrating comprehensive Two-Factor Authentication (2FA) functionality. This entails modifying the login page to include a second step for 2FA verification, employing Time-Based One-Time Password (TOTP) with backup tokens as fallback options, and allowing for configurable SMS-based OTP by deployments. Backend support for TOTP authentication will be seamlessly integrated using compatible libraries such as pyotp, ensuring smooth and secure 2FA functionality. Users will have the flexibility to enroll in 2FA via their account settings, with the chosen method verified during enrollment and a unique secret key generated for each user. Fallback options like backup codes or SMS/email-based OTP will be provided to address instances of 2FA device loss, ensuring uninterrupted access to user accounts. Additionally, security measures such as secure storage of TOTP secret keys and implementation of rate limiting on 2FA attempts will be enforced to mitigate the risk of unauthorised access and brute force attacks. Clear instructions and support will be offered to assist users in setting up and effectively using 2FA, along with the provision of backup codes for contingency purposes. Overall, this project represents a comprehensive approach to enhancing login security and user authentication within our platform.

For example, if a password is hacked, guessed, or even phished, 2FA requires a second factor such as a code to access the account. 2FA is essential to web security and protects against many common types of cyber threats, including stolen passwords.

FEATURES:

- Centralised Capacity Management: Efficiently allocates resources across hospitals, labs, and treatment centres.
- Patient Management System: Streamlines workflows for pandemic management and enables remote TeleICU services.
- Smart Dashboards: Provides district administration with a comprehensive view of the healthcare system.
- Digitization of Patient Records: Improves data accessibility and reduces errors associated with manual record-keeping.
- User-friendly Interface: Enhances overall user experience for healthcare providers and administrators.

Technical skill

- ➔ React.js
- ➔ Django

- ➔ Node.js
- ➔ HTML
- ➔ CSS
- ➔ MongoDB
- ➔ PostgreSQL
- ➔ Docker

Relevant Experiences

- **Demo**

  Explaining with code how my skill and experiences can contribute in the project:

  Step 1: Create a custom authentication backend (backend.py):

```
from django.contrib.auth.models import User
from django_otp import user_has_device

class OTPBackend:

    def authenticate(self, request, username=None, password=None, **kwargs):
        try:
            user = User.objects.get(username=username)
            if user.check_password(password):
                if user_has_device(user):
                    request.session['otp_user_id'] = user.id
                    return user
        except User.DoesNotExist:
            return None

    def get_user(self, user_id):
        try:
            return User.objects.get(pk=user_id)
        except User.DoesNotExist:
            return None
```

  Step 2: Create a view for generating and sending OTP:

```
From django_otp import devices_for_user
```

```
from django_otp.plugins.otp_static.models import StaticDevice


def send_otp(request):
    user = request.user
    for device in devices_for_user(user):
        if isinstance(device, StaticDevice):
            token = device.token_set.create()
            # Code to send OTP via SMS, email, etc.
            return HttpResponse("OTP sent successfully.")
    return HttpResponse("No OTP device found.")
```

Step 3: Create a view for verifying the OTP:

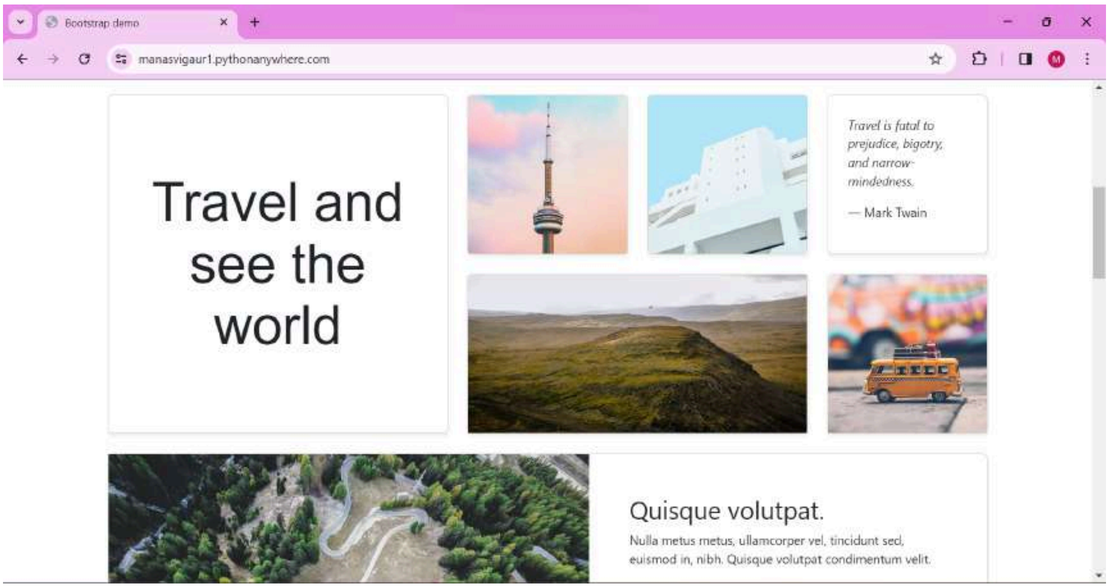
```
from django_otp import match_token
from django_otp.plugins.otp_static.models import StaticToken


def verify_otp(request):
    user_id = request.session.get('otp_user_id')
    user = User.objects.get(pk=user_id)
    token = request.POST.get('otp_token')
    if match_token(user, token):
        # OTP verification successful, login user
        login(request, user)
        return HttpResponse("OTP verified successfully.")

    return HttpResponse("OTP verification failed.")
```

Step 4: Setting up end Points for these views and activities:

```
urlpatterns = [

    ...

    path('send-otp/', send_otp, name='send_otp'),

    path('verify-otp/', verify_otp, name='verify_otp'),

]
```

- **TravelVista**

TravelVista is a travel website developed using Django and SQLite. It offers users a platform to explore travel destinations, plan trips, and book accommodations. The website's backend, powered by Django and SQLite, ensures fast and efficient data storage and retrieval. With its user-friendly interface and robust backend, TravelVista provides a seamless experience for travellers looking to plan their next adventure**.**



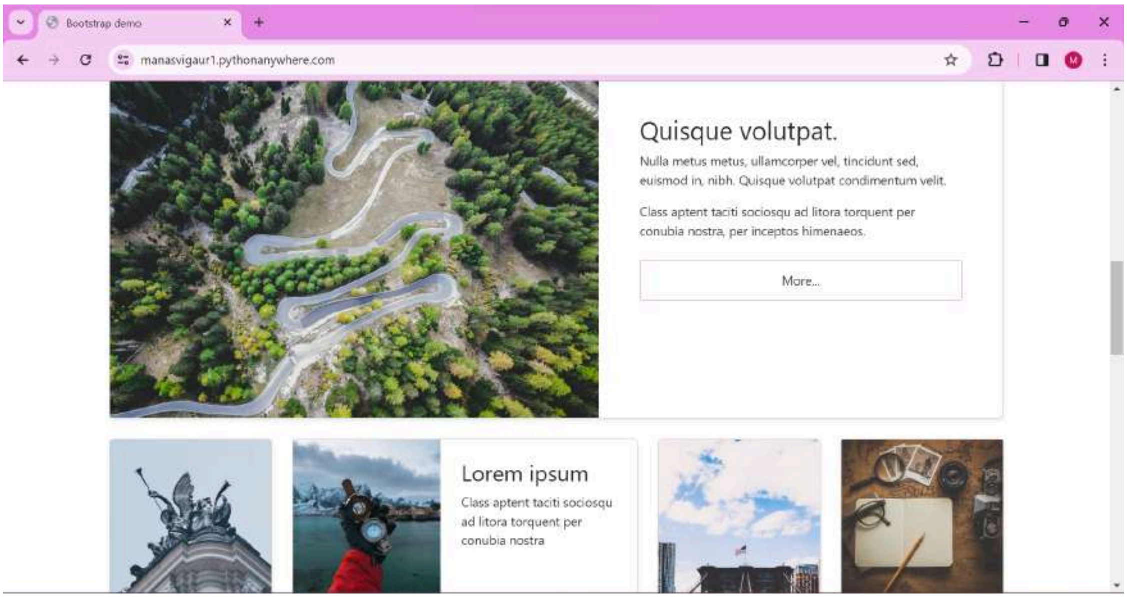
Travel and see the world

Travel is fatal to prejudice, bigotry, and narrow-mindedness.

— Mark Twain

Quisque volutpat.

Nulla metus metus, ullamcorper vel, tincidunt sed, euismod in, nibh. Quisque volutpat condimentum velit.



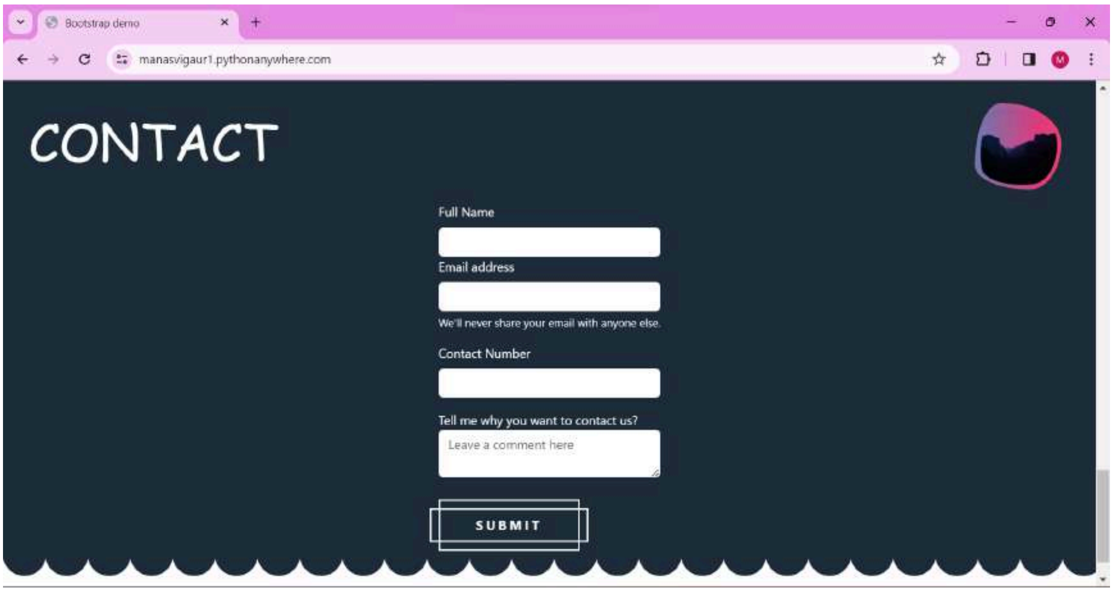
Quisque volutpat.

Nulla metus metus, ullamcorper vel, tincidunt sed, euismod in, nibh. Quisque volutpat condimentum velit.

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

Lorem ipsum

Class aptent taciti sociosqu ad litora torquent per conubia nostra



CONTACT

Full Name

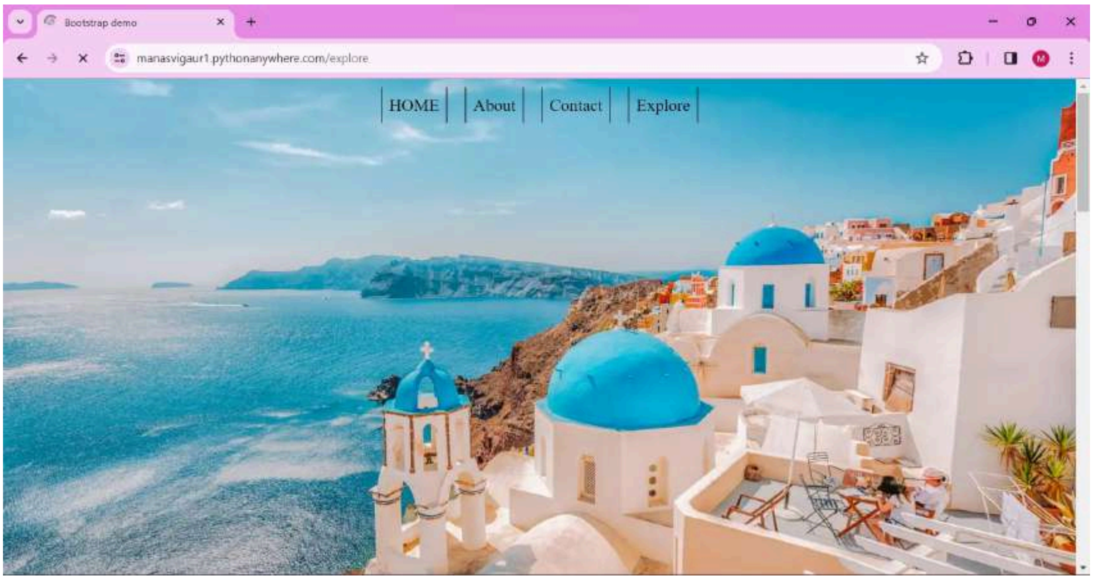
Email address

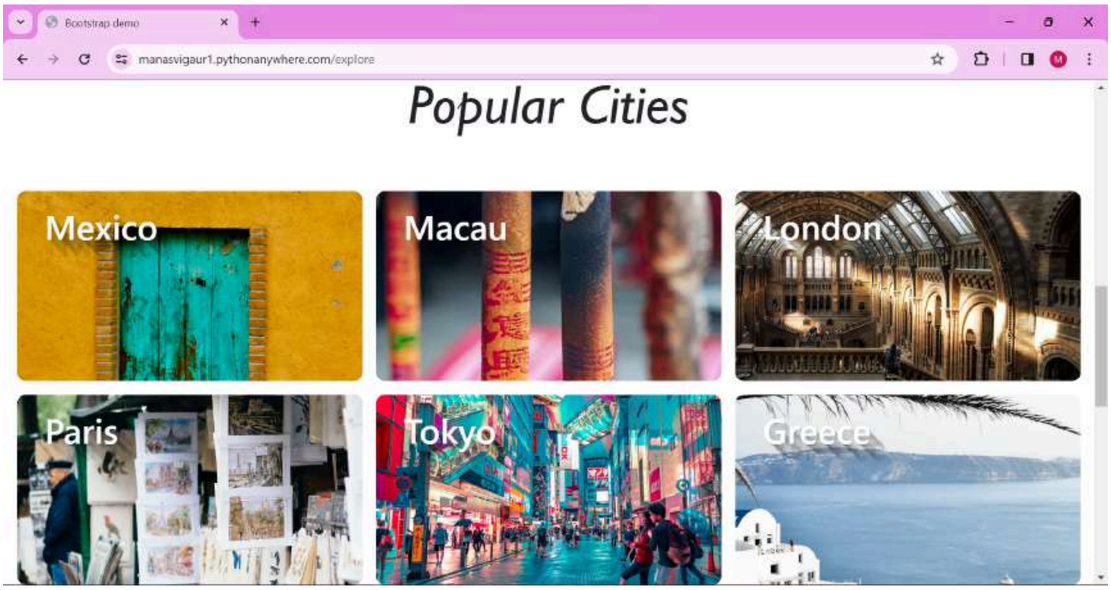
We'll never share your email with anyone else.

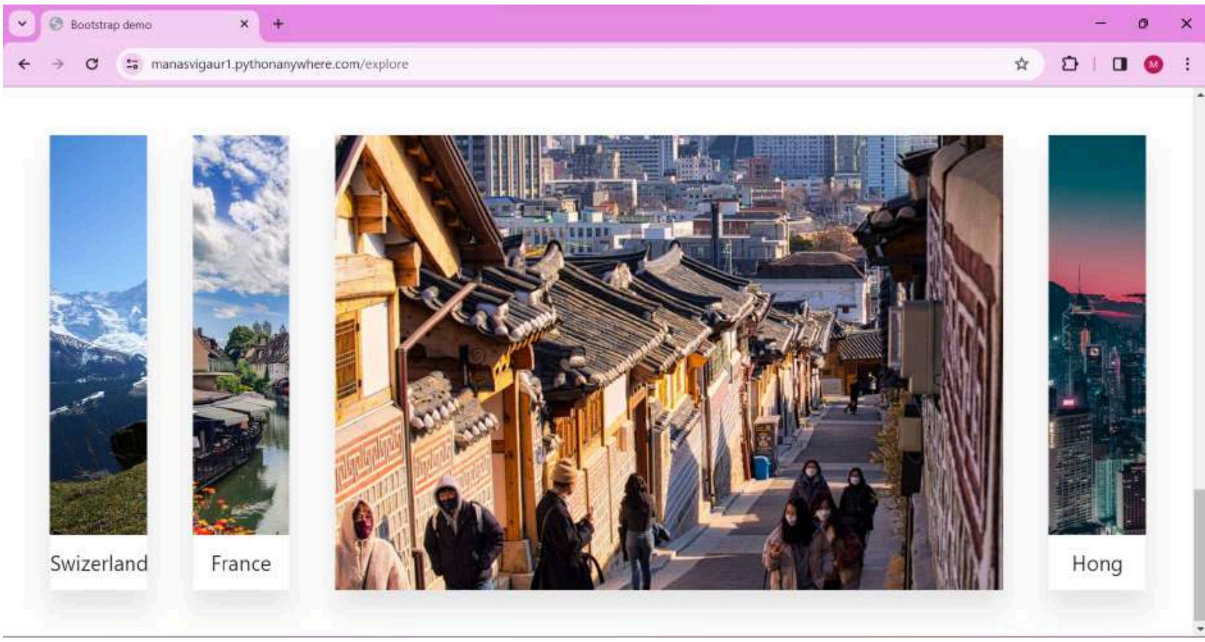
Contact Number

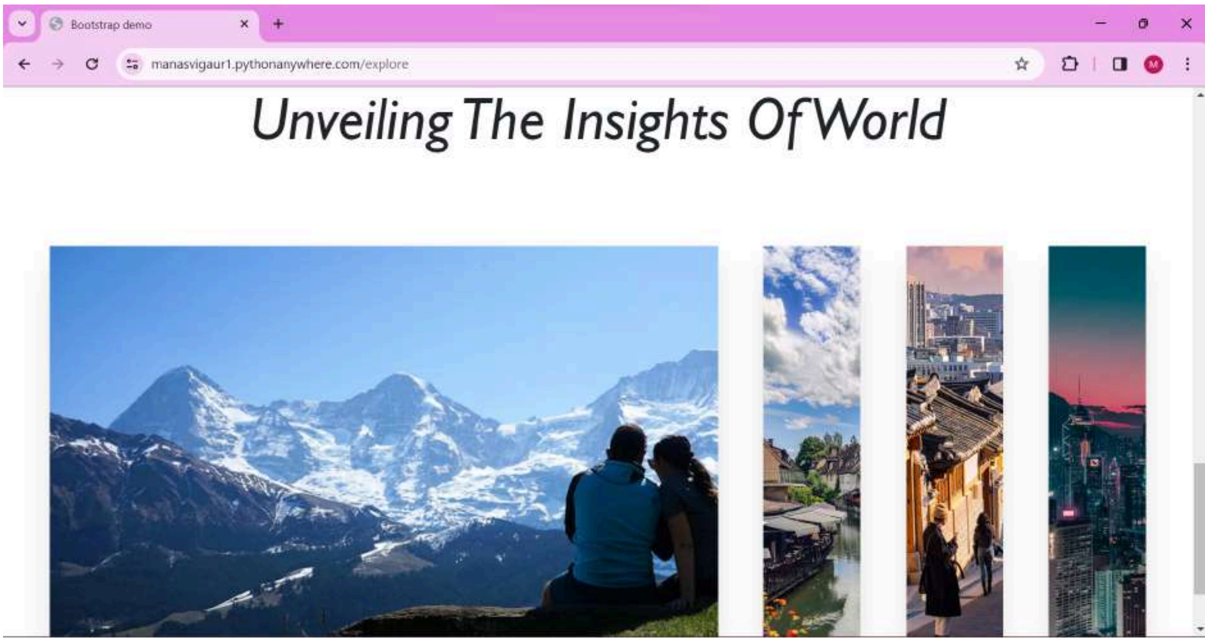
Tell me why you want to contact us?

Leave a comment here


Bootstrap demo
manasvigaur1.pythonanywhere.com/explore
HOME
About
Contact
Explore

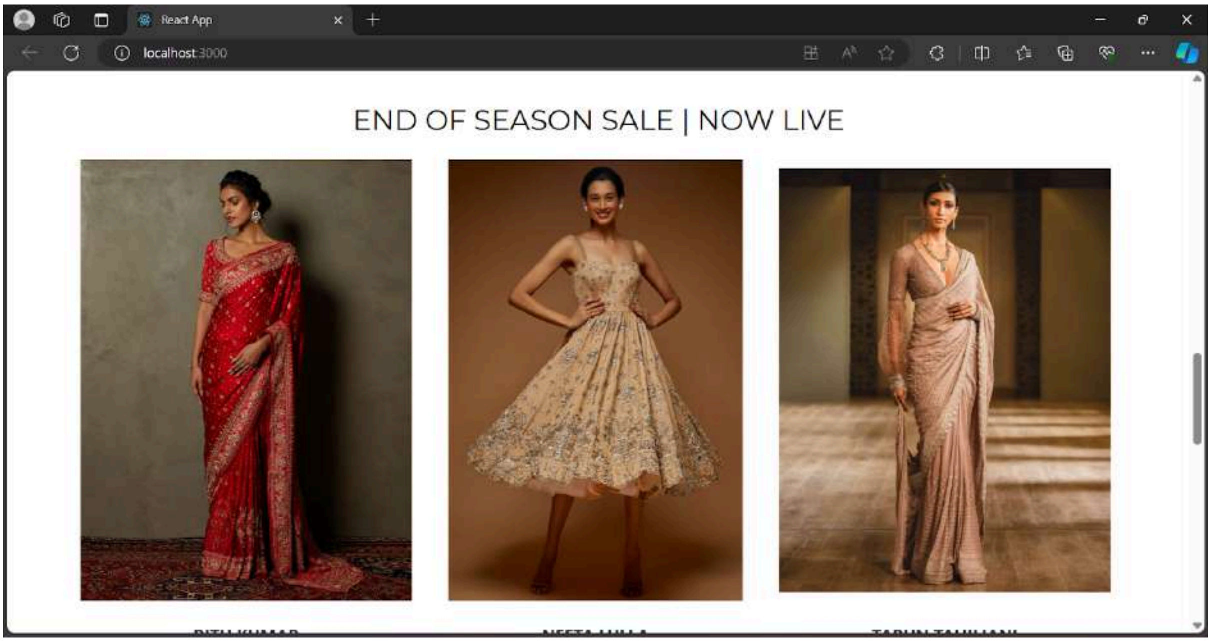

Bootstrap demo
manasvigaur1.pythonanywhere.com/explore
Popular Cities
Mexico
Macau
London
Paris
Tokyo
Greece




Bootstrap demo
manasvigaur1.pythonanywhere.com/explore
Unveiling The Insights Of World

- **BlissBuy**

  I have implemented an Ecommerce Website BlissBuy that integrates React for a user-friendly interface, MongoDB for a dynamic product catalogue, Django for secure authentication, and Node.js for efficient backend processing. With a responsive design, seamless checkout process, and real-time order tracking, BlissBuy offers a modern and convenient shopping experience for users.

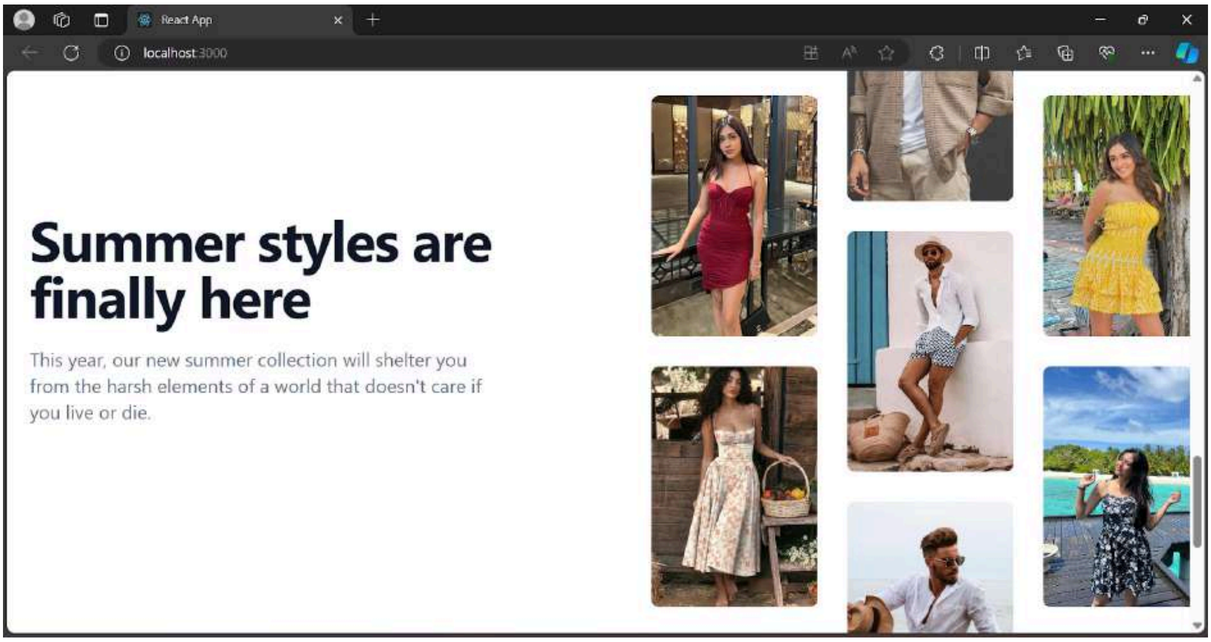


END OF SEASON SALE | NOW LIVE

RITU KUMAR

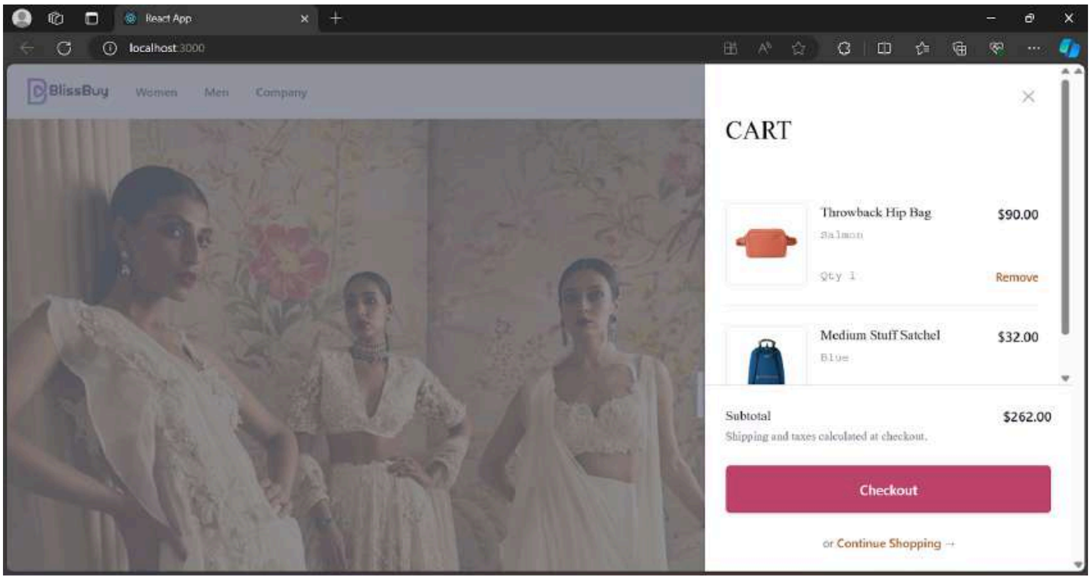
NEETA LULLA

TARUN TAHILIANI



Summer styles are finally here

This year, our new summer collection will shelter you from the harsh elements of a world that doesn't care if you live or die.

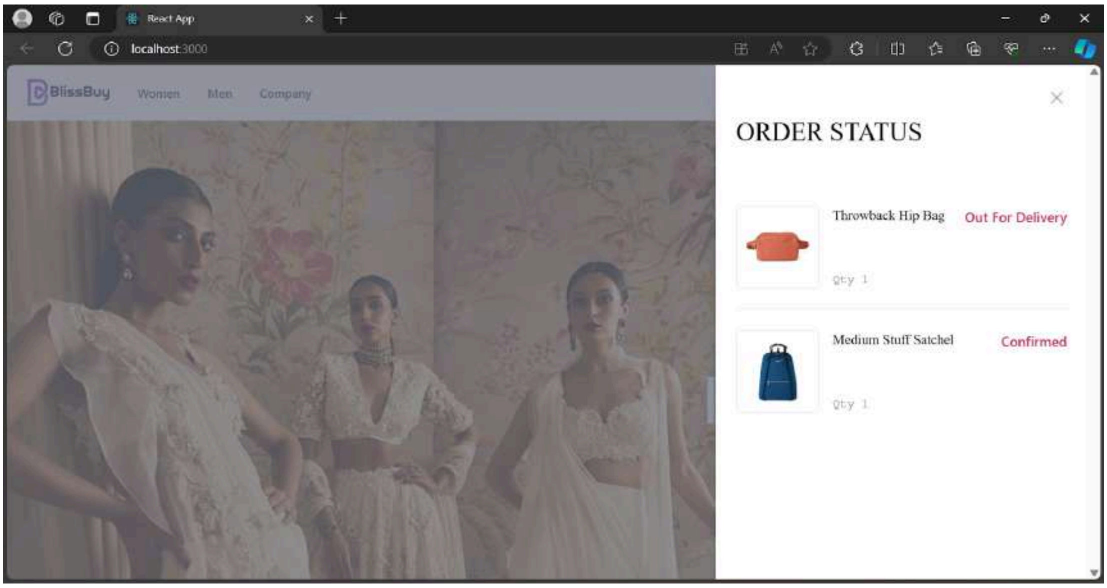


CART

Throwback Hip Bag $90.00
Salmon
Qty 1

Medium Stuff Satchel $32.00
Blue

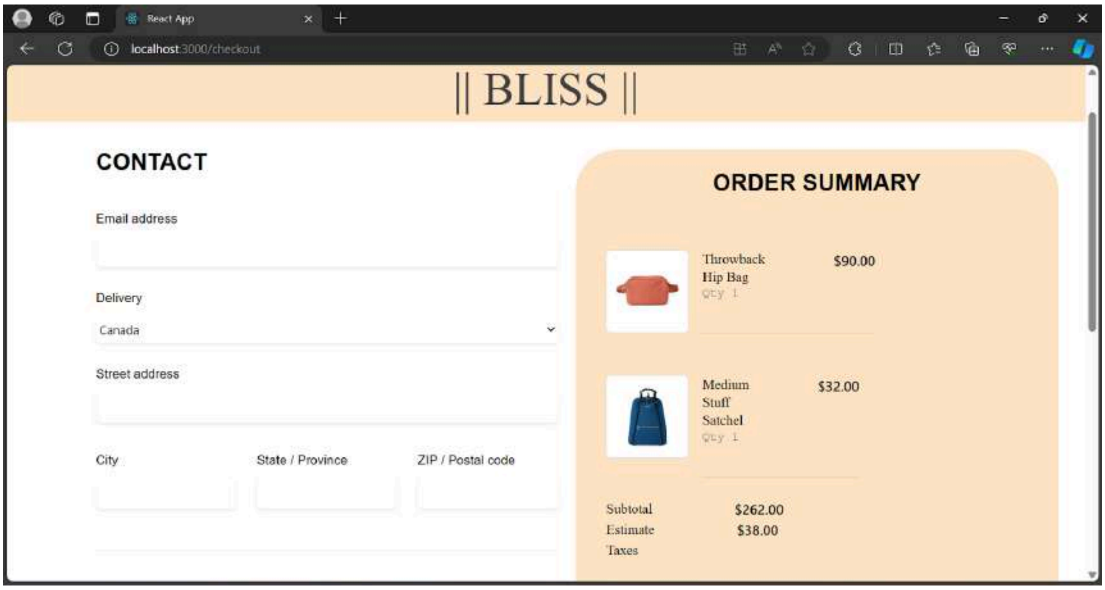
Subtotal $262.00
Shipping and taxes calculated at checkout.



ORDER STATUS

Throwback Hip Bag Out For Delivery
Qty 1

Medium Stuff Satchel Confirmed
Qty 1



|| BLISS ||

CONTACT

Email address

Delivery
Canada

Street address

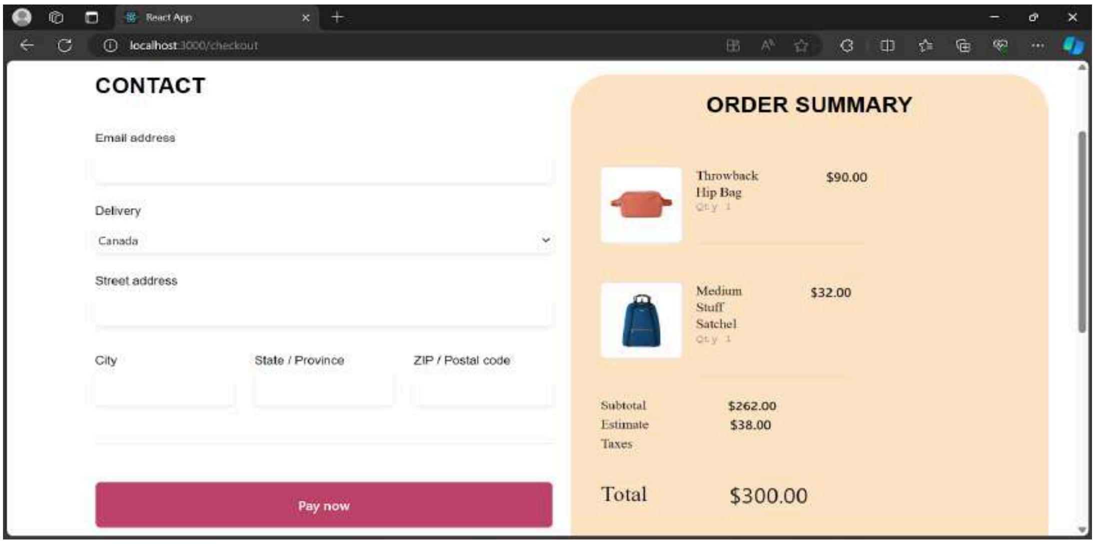
City State / Province ZIP / Postal code

ORDER SUMMARY

Throwback Hip Bag $90.00
Qty 1

Medium Stuff Satchel $32.00
Qty 1

Subtotal $262.00
Estimate Taxes $38.00



CONTACT

Email address

Delivery

Canada

Street address

City

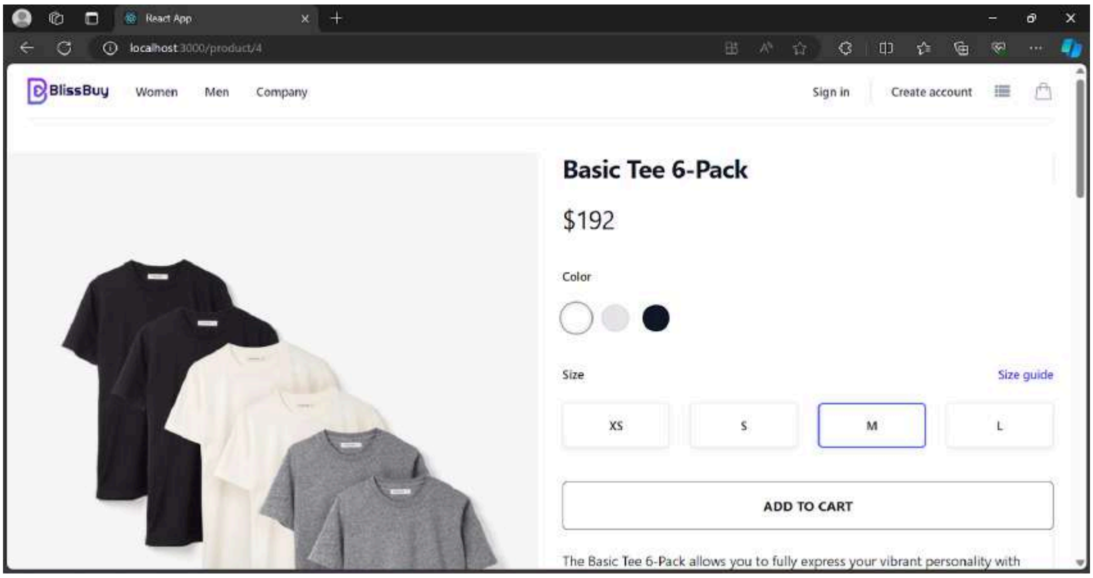
State / Province

ZIP / Postal code

Pay now

ORDER SUMMARY

Throwback Hip Bag Qty 1	$90.00
Medium Stuff Satchel Qty 1	$32.00
Subtotal	$262.00
Estimate Taxes	$38.00
Total	$300.00



Basic Tee 6-Pack

$192

Color

○ ○ ●

Size

Size guide

XS S M L

ADD TO CART

The Basic Tee 6-Pack allows you to fully express your vibrant personality with

Implementation Timeline and Milestone

May 17 - May 24 Community Bonding Period	Familiarise yourself with the CARE project and its current authentication system.I will be going through my University final exams (May 12- May 22). I would also like to have discussions regarding the different implementation ways to be used in creating a 2FA model to optimise implementation and start exploring those possibilities. Discuss the implementation plan with mentors and finalise the approach.Plan the implementation details for both frontend (ReactJS) and backend (Django) going through my University final exams.
May 25 - june 2 Backend Implementation	Set up the backend environment. Integrate the TOTP library (e.g., PyOTP) for generating and verifying TOTP codes. Implement backend support for user enrollment in 2FA and secure storage of TOTP secret keys.
June 3 - June 10 **Frontend Integration**	Traveling most of this week, Update the frontend login page to include a field for entering the verification code.
June 11 - June 18	Implement frontend logic to handle 2FA enrollment and verification processes. Test the frontend integration with the backend for 2FA functionality.
June 19 - June 26 **Backup Code Provision and Security Measures**	Implement backup code provision and allow users to access their accounts using backup codes.
June 27 - July 4	Implement security measures such as rate limiting on 2FA attempts to mitigate brute force attacks. Test the backup code provision and security measures.

July 5 - July12 **User Support and Guidance**	Provide clear instructions and support for users to set up and use 2FA. Ensure users can easily add or remove 2FA from their profile settings. Test the user support and guidance features.
July 13 - July 20 **QA and Code Review**	Conduct QA testing to ensure all 2FA features work as expected. Review the code for logic and structure, ensuring it is sustainable for long-term project use. Finalise documentation and prepare for code submission.
July 21 - July 28 **Final Testing and Optimization**	Finalise documentation and prepare for code submission.
July 29 - Aug 5	Perform final testing of all features and functionalities. Optimise the code and improve the user experience where necessary.
Aug 6 - Aug 13 **Project Review and Completion**	Present the implemented 2FA features to the mentors for review.
Aug 14- Aug 21	Address any feedback or issues raised during the review.
Aug 22 - Aug 29	Complete the implementation and ensure the project meets all acceptance criteria.
Aug 30 - Sep 6	*Submitting the final code*
Sep 7 - Sep 14	Wrap up

Sept 15 - Sept 22	GSoC Report

Summary About Me

I'm Manasvi Gaur, a seasoned software developer specialising in a wide array of technologies. With a keen focus on front-end development, I excel in crafting captivating user interfaces using React.js, ensuring optimal user experience across devices.

My proficiency extends to back-end development, where I leverage Node.js and Express.js to build robust, scalable server-side applications. I have a strong command of database management, utilising MongoDB for its flexibility and PostgreSQL for its reliability in handling relational data.

Additionally, I am well-versed in Django, a powerful Python framework, which I use to create secure and efficient web applications. My expertise also includes mastering web technologies such as HTML, CSS, and JavaScript, along with TypeScript for enhanced development workflows.

I have a solid grasp of containerization with Docker, enabling me to streamline the deployment and management of applications. Furthermore, my proficiency in Android development allows me to create seamless mobile experiences using Java .

With a proven track record of delivering successful projects, I actively contribute to open source to stay abreast of the latest trends and technologies. I am deeply passionate about software development and continuously seek opportunities to expand my skills and knowledge in the field.

Why Care?

My motivation for applying to the CARE project stems from a deep-seated belief in the transformative power of technology in revolutionising healthcare management. The CARE project's goal of centralising capacity management, patient management, and digitising patient records aligns perfectly with my passion for developing innovative solutions that improve people's lives. I have solved issues and become familiar with codebase
I aim to bring my extensive experience and expertise in ReactJS, TypeScript, Django, and Python to the CARE project. I am committed to enhancing the security of the login process through the implementation of a robust Two-Factor Authentication (2FA) mechanism. My goal is to contribute to the project's success by delivering high-quality, secure, and user-friendly authentication features that align with the project's objectives and exceed user expectations.

Achievements and Academies:

- I participated in SIH 2023 , WE by google and flipkart runway
- I scored CGPA (till now) in academy: 9.18

Availability

I am available for full time work in this summer because I have no vacation plan other than this project. In the official GSoC period, I will devote the sufficient time per week which is required for this project. I will remain in contact in all time through the e-mail and on any other communication channel with the community.
During the community bonding period, I have exams which will get over on the 22th of May. I do not yet know the exact date when my university will reopen. However, if a little amount of work is still remaining, I can work from the university after my classes for 4-5 hours every day until the project is complete.

Other Commitments

- **Do you have any other commitments during the GSoC period?**
  No, I am full time available to work with Google Summer of Code.

- **Have you applied to any other organizations? If so, to whom and do you have a preferred project/org?**
  No,i am totally focused on care

Open Source Contributions {OHC Repo}

- Solved File Management Exceptions #7390 {Merged} - https://github.com/coronasafe/care_fe/pull/7390
- [Enhancement] Ventilator Parameters Input - Validate Consultation Bed for Linked Ventilator Asset #7397 {In Progress} - https://github.com/coronasafe/care_fe/issues/7397