Frontend

To make simple react frontend

git clone --single-branch -b react-mini <https://github.com/safak/youtube.git>

**LIBRARIES:**

Yarn add styled-components

Yarn add @material-ui/core

**Using Urbanist Font family:**

Fonts.google.com

Urbanist

Select all styles

Copy link from right side and also font family

Update index.html in public folder as;

<link href="https://fonts.googleapis.com/css2?family=Urbanist:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">

  <style>

    \* {

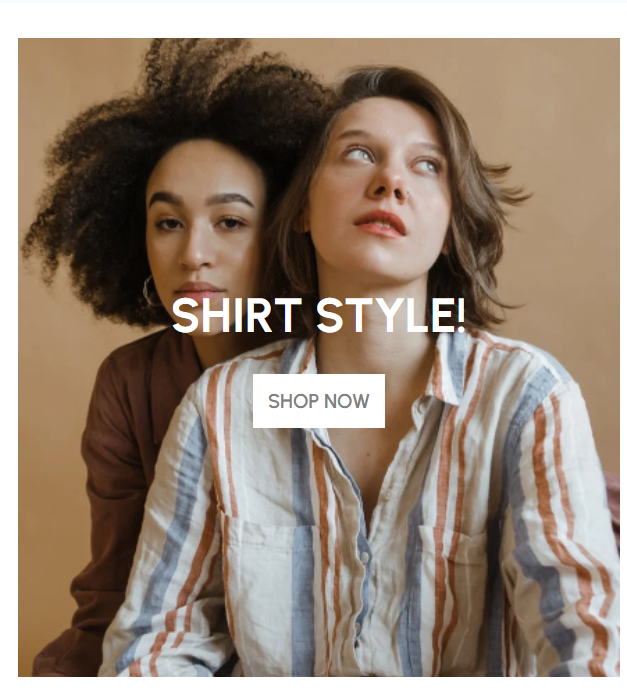
      margin: 0;

      font-family: 'Urbanist', sans-serif;

    }

  </style>

**Text on Image:**



For text and button section, make it in a div

Div should be absolute and

const Container = styled.div`

    flex: 1;

    margin: 3px;

    height: 70vh;

    position: relative;

`

const Info = styled.div`

    position: absolute;

    width: 100%;

    height: 100%;

    top: 0;

    left: 0;

    display: flex;

    align-items: center;

    justify-content: center;

    flex-direction: column;

`

<Container>

        <Image src={item.img}/>

        <Info>

            <Title>{item.title}</Title>

            <Button>SHOP NOW</Button>

        </Info>

    </Container>

It’s complete code is

import React from 'react'

import styled from 'styled-components'

const Container = styled.div`

    flex: 1;

    margin: 3px;

    height: 70vh;

    position: relative;

`

const Image = styled.img`

    width: 100%;

    height: 100%;

    object-fit: cover;

`

const Info = styled.div`

    position: absolute;

    width: 100%;

    height: 100%;

    top: 0;

    left: 0;

    display: flex;

    align-items: center;

    justify-content: center;

    flex-direction: column;

`

const Title = styled.h1`

    color: white;

    margin-bottom: 20px;

`

const Button = styled.button`

    border: none;

    padding: 10px;

    background-color: white;

    color: gray;

    cursor: pointer;

    font-weight: 600;

`

function CategoryItem({item}) {

  return (

    <Container>

        <Image src={item.img}/>

        <Info>

            <Title>{item.title}</Title>

            <Button>SHOP NOW</Button>

        </Info>

    </Container>

  )

}

export default CategoryItem

**USING STYLED COMPONENTS:**

Yarn add styled-components

import React from 'react'

import styled from 'styled-components'

const Container = styled.div`

    height: 60px;

    background-color: black;

`

function Navbar() {

  return (

    <Container>

        Navbar

    </Container>

  )

}

export default Navbar

We create container styled component targeting div, and instead of div in html, we used container

**To divide 3 columns or divs or div in equal spaces:**

const Left = styled.div`

    flex: 1;

`

const Center = styled.div`

    flex: 1;

`

const Right = styled.div`

    flex: 1;

`

Use of Props in Styled components:

const Arrow = styled.div`

    width: 50px;

    height: 50px;

    background-color: #fff7f7;

    border-radius: 50%;

    display: flex;

    align-items: center;

    justify-content: center;

    position: absolute;

    top: 0;

    bottom: 0;

    left: ${props => props.direction === "left" && "10px"};

    right: ${props => props.direction === "right" && "10px"};

    margin: auto;

`

function Slider() {

  return (

    <Container>

        <Arrow direction="left">

            <ArrowLeftOutlined/>

        </Arrow>

        <Arrow direction="right">

            <ArrowRightOutlined/>

        </Arrow>

HOVER IN STYLED COMPONENTS:

const Icon = styled.div`

    width: 40px;

    height: 40px;

    border-radius: 50%;

    background-color: white;

    display: flex;

    align-items: center;

    justify-content: center;

    margin: 10px;

    cursor: pointer;

    transition: all 0.5s ease;

    &:hover{

        background-color: #e9f5f5;

        transform: scale(1.1);

    }

IMPACT OTHER CLASS ON HOVER:

Change info class when hover on container:

const Info = styled.div`

    opacity: 0;

    width: 100%;

    height: 100%;

    position: absolute;

    top: 0;

    left: 0;

    background-color: rgba(0,0,0,0.2);

    z-index: 3;

    display: flex;

    align-items: center;

    justify-content: center;

    transition: all 0.5s ease;

    cursor: pointer;

`

const Container = styled.div`

    flex: 1;

    margin: 5px;

    min-width: 280px;

    height: 350px;

    display: flex;

    align-items: center;

    justify-content: center;

    background-color: #f5fbfd;

    position: relative;

    &:hover ${Info}{

        opacity: 1;

    }

`

MOBILE RESPONSIVENESS IN STYLED COMPONENTS

First Method:

const Container = styled.div`

    height: 60px;

@media only screen and (max-width:480px){

Height: 40px;

}

   `

Second Method:

Create JS function in responsive.js file at src destination:

import {css} from 'styled-components'

export const mobile  = (props) => {

    return css`

        @media only screen and (max-width: 480px) {

            ${props}

        }

    `

};

Import where media query required

import {mobile} from '../responsive'

const Container = styled.div`

    height: 60px;

    ${mobile( {backgroundColor: "red"})}

`

Backend

Index.js

Npm init

Install libraries

Node index.

Yarn add mongoose nodemon dotenv express

Yarn add crypto-js (password hashing)

yarn add jsonwebtoken (for verify whether order etc belongs to corresponding user)

yarn add stripe

**STRIPE PAYMENT METHOD:**

Backend:

const router = require("express").Router();

// const stripe = require("stripe")(process.env.STRIPE\_KEY);

const KEY = process.env.STRIPE\_KEY

const stripe = require("stripe")(KEY);

router.post("/payment", (req, res) => {

  stripe.charges.create(

    {

      source: req.body.tokenId,

      amount: req.body.amount,

      currency: "usd",

    },

    (stripeErr, stripeRes) => {

      if (stripeErr) {

        res.status(500).json(stripeErr);

      } else {

        res.status(200).json(stripeRes);

      }

    }

  );

});

module.exports = router;

And add stripe secret key (stripe sign in -> developers -> api keys -> copy secret key)in .env file

MONGO\_URL = mongodb+srv://admin:DscVShSuT1ywR5gz@cluster0.vk4fa.mongodb.net/?retryWrites=true&w=majority

PORT = 5000

PASS\_SEC = muneeb

JWT\_SEC = muneeb

STRIPE\_KEY = sk\_test\_51JhERSGkYc9g1BvdKp9E0XKpPfoSNDOjmj1SfTjSEhCoARvnO40vktSAlo8xH2UlwYzl76LkcZPaA8e9mqn89Q7u00Ix1u8hFw

HASHING PASSWORD:

router.post('/register', (req,res)=>{

    const newUser = new User({

        username: req.body.username,

        email: req.body.email,

        password: CryptoJS.AES.encrypt(req.body.password, process.env.PASS\_SEC).toString(),

    })

    newUser

    .save()

    .then(result=>{

        res.status(201).json(result)

    })

    .catch(err=>{

        res.status(500).json(err)

    })

})

PASS\_SEC passed in .env file

Admin