DressHow

A react based web application

Aksa
228491
BSCS 7B

Problem Statement

Every clothing brand has its own website. Therefore, to find new clothes and related items, we have to search through a number of different websites to find the item we like. Most brands have same type of clothes and varying prices. Therefore, it becomes difficult for us to find the clothes according to our choice and budget range.

Proposed Solution

DressHow is a react based web-application that provides all the items from popular brands at one place. Users can easily compare items from different brands according to their prices and thus make a better choice.

Main features

- Places all the new arrival items from the brands such as khaadi, outfitters and breakout.
- Filter out the items according to the brand(s) of your choice.
- Find the items according to their budget range
  - Sort the items according to the lowest or highest price.
- Wanna buy the item? One step away
  - Clicking on the items directs to the official page
- Price Analysis
  - The website contains graphs comparing the item prices from different brands
  - Shows minimum, maximum and average price of each brand as well
- Provide feedback to analyze and improve user experience

Tools used

Back-end

- Express
- Puppeteer-Nodejs library
- Mongoose

Front-end

- React
- Materialize for styling
- Chart.js for displaying charts-javascript library

Back-end

- The information of the items is extracted using puppeteer (a nodejs library)
  - Information includes image url, name , price and the official page url of the item.
  - Data is collected asynchronously from all the brands using css selectors
  - Methods such as page.click(),document.querySelectorAll(), page.evaluate(), page.goto() etc. are used
  - Data is stored in json format.
- Express.js is used to send the response and receive the request.
- Mongoose is used to store the feedback form information.

Backend-Main server.js File

```
var express = require("express");

const formController = require('./controller/formController')
const scrapingController = require('./controller/scrapingController'

var bodyParser = require("body-parser")
var multer = require('multer')
var upload = multer()
var app = express();
app.use(bodyParser.json())
app.use(bodyParser.urlencoded({extended:true}))
app.use(upload.array())


app.use('/api/items', scrapingController);
app.use('/api/form', formController);

const cors = require('cors');
app.use(cors())
const port =  5000;
app.listen(port, ()=>console.log('server started on '+ port))
```

server.js file

```
router.get("/", function(req, res)
{
  var data1 = data;
  var random_array = shuffle(data1,{copy:true});
  console.log(random_array.length);
  console.log(random_array.length);
  return res.send(random_array);
})
```

Router.get() method in scrapingController

ScrapingController
Extracting items from khaadi

```
// get the khaadi clothes detail
;(async () => {
    console.log("khaadi")
    const browser = await puppeteer.launch({headless:false});
    const page = await browser.newPage();
    await page.goto("https://www.khaadi.com/pk/new-in/woman.html",{waitUntil:'networkidle2', timeout:0});
    var results= {li:[],image:[],name:[],price:[]};
    var lastPageNumber = 3;

    for (let index = 0;index<lastPageNumber;index++){
      await page.waitFor(10000);
      var result = await extractKhaadiItems(page);
      results.li = results.li.concat(result.li);
      results.image = results.image.concat(result.image);
      results.name = results.name.concat(result.name);
      results.price= results.price.concat(result.price);
      if (index!=lastPageNumber-1){
        await page.click('div.toolbar-bottom div.pager-html div.arrow-pagination a.action.next');
      }
    }
    await browser.close();
    for (var i=0; i< results.li.length;i++)
    {
      var pr = parseInt(results.price[i].replace(/[^0-9]/g,""));
      var item = {id:k, link:results.li[i], img:results.image[i], name:results.name[i], price:pr, brand:"khaadi"};
      data.push(item);
      k++;
    }
})();
```

```
async function extractKhaadiItems(page)
{

 await page.waitForSelector("div.product-item-info");
 await page.waitForSelector("div.product.details.product-item-details");
 var temp = await page.evaluate(()=>{
    var link = Array.from(document.querySelectorAll("div.product-item-info a.product.photo.product-item-photo")).map(val=>val.href);
    var img = Array.from(document.querySelectorAll
    ("div.products.catlog-products.wrapper.grid.products-grid div.product-item-info span.product-image-wrapper-front img.product-image-photo")).map(val=>val.ge
    var name= Array.from(document.querySelectorAll("div.product.details.product-item-details strong.product.name.product-item-name a.product-item-link"))
    .map(val=>val.innerText);
    var price= Array.from(document.querySelectorAll("div.product.details.product-item-details div.price-box.price-final_price span.price"))
    .map(val=>val.innerText);
    return {li: link, image:img, name:name, price:price}
 })
 return temp;
}
```

Extracting items from Outfitters

```
;(async()=>{
  console.log("outfitters")
  const browser = await puppeteer.launch({headless:false});
    const page = await browser.newPage();
    await page.goto("https://outfitters.com.pk/collections/womenwinter19",{waitUntil:'networkidle2', timeout:0});
    await page.waitForSelector("div.inner.product-item div.inner-top");
    var temp = await page.evaluate(()=>{
        var link = Array.from(document.querySelectorAll("div.inner.product-item div.inner-top div.product-top div.product-image a.product-grid-image"))
        .map(val=>val.href);
        var img = Array.from(document.querySelectorAll("div.inner.product-item div.inner-top div.product-top div.product-image a.product-grid-image>img"))
        .map(val=>val.src);
        var name = Array.from(document.querySelectorAll("div.inner.product-item div.inner-top div.product-bottom a.product-title.pull-left"))
        .map(val=>val.innerText);
        var price = Array.from(document.querySelectorAll("div.inner.product-item div.inner-top div.product-bottom div.price-box span.money"))
        .map(val=>val.innerText);
        return {li:link, image:img, name:name, price:price, brand:"outfitters"}
      })
    await browser.close();
     for (var i=0; i< temp.li.length;i++)
    {
      var pr = parseInt(temp.price[i].replace(/[^0-9]/g,""));
      var item = {id:k, link:temp.li[i], img:temp.image[i], name:temp.name[i], price:pr, brand:temp.brand};
      data.push(item);
      k++;
    }

})();
```

Extracting items from Breakout

```
;(async()=>{
  console.log("breakout")
  const browser = await puppeteer.launch({headless:false});
    const page = await browser.newPage();
    await page.goto("https://www.breakout.com.pk/new-in-2",{waitUntil:'networkidle2', timeout:0});
    await page.waitForSelector("div.item-grid div.item-box div.product-item");
    for (var m =0; m<8;m++)
    {
    var temp = await page.evaluate(()=>{
        var link = Array.from(document.querySelectorAll("div.item-grid div.item-box div.product-item div.picture a")).map(val=>val.href);
        var img = Array.from(document.querySelectorAll("div.item-grid div.item-box div.product-item div.picture a>img")).map(val=>val.src);
        var name = Array.from(document.querySelectorAll("div.item-grid div.item-box div.product-item div.details h2.product-title a"))
        .map(val=>val.innerText);
        var price = Array.from(document.querySelectorAll
        ("div.item-grid div.item-box div.product-item div.details div.add-info div.prices span.price.actual-price")).map(val=>val.innerText);
        return {li:link, image:img, name:name, price:price, brand:"breakout"}
      })
    await scrollToBottom(page);
    await page.waitFor(10000);
  }
    await browser.close();
    for (var i=0; i< temp.li.length;i++)
    {
      var pr = parseInt(temp.price[i].replace(/[^0-9]/g,""));
      var item = {id:k, link:temp.li[i], img:temp.image[i], name:temp.name[i], price:pr, brand:temp.brand};
      data.push(item);
      k++;
    }

})();

async function scrollToBottom(page) {

  const distance = 100; // should be less than or equal to window.innerHeight
  const delay = 100;
  while (await page.evaluate(() => document.scrollingElement.scrollTop + window.innerHeight < document.scrollingElement.scrollHeight)) {
    await page.evaluate((y) => { document.scrollingElement.scrollBy(0, y); }, distance);
    await page.waitFor(delay);
  }
}
```

formController

```
const express =  require('express');
var router = express.Router();
var bodyParser = require("body-parser")
var multer = require('multer')
var upload = multer()
require("../models/db");
var Form = require('../models/form.model')


router.post("/",function(req, res)
{
     console.log(req.body)
     var name = req.body.name;
     var phone = req.body.phone;
     var email= req.body.email;
     var rate = req.body.rate;
     var comment= req.body.comment;
     var newReview = {
          name:name,
          phone:phone,
          email:email,
          rate:rate,
          comment:comment,
     }
     var review = new Form(newReview)
     review.save(function(err,Form)
     {
          if (err) throw err;
          console.log("review saved in database")
          console.log(Form)
     })
     res.redirect('../contact')

})


module.exports = router;
```

Front-end
App.js and Router.js

```jsx
import React, {Component} from 'react';
import Routes from './components/Routes.js';
import './App.css';

class App extends Component {
  constructor()
  {
    super()
  }
  render()
  {
    return (
      <div>
        <Routes />
      </div>
  );
  }

}

export default App;
```

```jsx
import React, {Component} from 'react';
import Home from './Home.js';
import Graph from './Graph.js';
import Contact from './Contact.js';
import Navbar from './Navbar.js';
import Notfound from './Notfound.js'
import {BrowserRouter as Router, Route, Switch} from 'react-router-dom';

class Routes extends Component {
  constructor(props)
  {
    super(props)
  }
  render()
  {
    return (

      <Router>
          <Switch>
            <Route exact path = "/" component={Home} />
            <Route exact path = "/home" component={Home} />
            <Route exact path = "/graph" component={Graph} />
            <Route exact path = "/contact" component={Contact} />
            <Route path ="*" component={Notfound} />
          </Switch>
      </Router>

  );
  }

}

export default Routes;
```

Index/ Home component

```
import React, {Component} from 'react';
import Navbar from './Navbar.js';
import Items from './Items.js';
import Pagination from './Pagination.js'
import style from './style/welcome.module.css';
import Category from './Category.js';
import Sort from './Sort.js';
import Footer from './Footer.js';
const axios = require('axios').default;
export default class Home extends Component{
  constructor(props){
    super(props);
    this.state = {
      items:[],
      filteredItems:[],
      loading:false,
      currentPage:1,
      itemsPerPage:12,
    }
  }
```

```
componentDidMount()
{

 const fetchPosts = async () => {
   this.setState({loading:true})
   await axios.get('/api/items').then(response=>
   this.setState({items:response.data},()=>console.log("products fetched", response.data)))
   this.setState({loading:false})
   this.setState({filteredItems:this.state.items});
 };


fetchPosts();
}
```

Fetch items from back-end

Display items
Home component

Pagination is used to display only 12 items per page

```
// Change page
const paginate = pageNumber => this.setState({currentPage:pageNumber});
//category
```

```
const indexOfLastPost = this.state.currentPage * this.state.itemsPerPage;
const indexOfFirstPost = indexOfLastPost - this.state.itemsPerPage;

var array = this.state.filteredItems;
var currentPosts = array.slice(indexOfFirstPost, indexOfLastPost);
```

```
<Items items={currentPosts} loading={this.state.loading} />
    <Pagination
      postsPerPage={this.state.itemsPerPage}
      totalPosts={this.state.filteredItems.length}
      currentPage = {this.state.currentPage}
      paginate={paginate}
    />
```

Items component

```
import React from 'react';
import style from './style/items.module.css';
const Items = ({ items, loading }) => {
  if (loading) {
    return <h2>Loading...</h2>;
  }
 for (let m=0;m<items.length;m++)
      console.log("id:"+items[m].id+" brand:"+items[m].brand)

  return (
      <div>
       <div className="row container">
      {items.map(item=>(
        <div className="col s6 m6 l4" key={items.id}>
          <div className={style.item+" card-action hoverable"}><a href={item.link} target="_blank"><div className="card">
            <div className="card-image">
              <img src={item.img} width="400px" height="600px" />
              <span className={style.itemname + " card-title valign-wrapper"} >{item.name}</span>
            </div>
            <div className={style.itemprice+" card-content"}>
              PKR {item.price}
            </div>
            <div className={style.itembrand + " card-content"}>
              {item.brand}
            </div>
          </div></a>
            </div>
        </div>
      ))}</div>
      </div>
  );
};

export default Items;
```

Pagination Component

```
import React, {Component} from 'react';
import styles from 'materialize-css';

const Pagination = ({ postsPerPage, totalPosts, currentPage, paginate }) => {
  const pageNumbers = [];

  for (let i = 1; i <= Math.ceil(totalPosts / postsPerPage); i++) {
    pageNumbers.push(i);
  }
  return (
      <ul className="pagination container center">
        {currentPage == 1 ?<li class="disabled" style={{pointerEvents:"none"}}>
        <a href="#!"><i class="material-icons">chevron_left</i></a></li>:<li class="waves-effect"><a onClick ={() => paginate(currentPage-1)} href='#'>
        <i class="material-icons">chevron_left</i></a></li>
        }
        {pageNumbers.map(number => (
          currentPage == number ? <li key={number} className="active waves-effect">
            <a onClick ={() => paginate(number)} href='#'>
              {number}
            </a>
          </li> : <li key={number} className="waves-effect">
            <a onClick ={() => paginate(number)} href='#'>
              {number}
            </a>
          </li>
        ))}
        {currentPage == pageNumbers[pageNumbers.length-1] ?<li class="disabled">
        <a href="#!"><i class="material-icons">chevron_right</i></a></li>:<li class="waves-effect"><a onClick ={() => paginate(currentPage+1)} href='#'>
        <i class="material-icons">chevron_right</i></a></li>
        }
      </ul>
  );
};

export default Pagination;
```

Filter items according to specific brand

```
export default class Category extends Component{
    constructor(props){
        super(props);
        this.state = {
        }
    }
    render(){
        console.log(this.props)
        return(
            <nav className={style.category + " container"}>
                <div className="nav-wrapper">
                  <ul>
                    <li style={{marginLeft:'20px'}}><a className={style.items} onClick={()=>this.props.filter("all")}  href="#">All</a></li>
                    <li><a className={style.items} onClick={()=>this.props.filter("khaadi")}  href="#">Khaadi</a></li>
                    <li><a className={style.items} onClick={()=>this.props.filter("outfitters")} href="#">Outfitters</a></li>
                    <li><a className={style.items} onClick={()=>this.props.filter("breakout")} href="#">Breakout</a></li>
                  </ul>
                </div>
              </nav>

        )
    }
}
```

Category Component

```
const filter = (category)=> {
  if (category == 'all')
  {
    this.setState({filteredItems:this.state.items})
  }
  else
  {
  let filt_array = this.state.items.filter(item => item.brand == category)
  this.setState({filteredItems:filt_array})
  }
};
```

In Home Component

Sort according to price

```jsx
import React, {Component} from 'react';
import styles from 'materialize-css';
import style from './style/sort.module.css';

export default class Sort extends Component{
    constructor(props){
        super(props);
        this.state = {
        }
    }
    render(){
        return(
            <div className="container center">
            <a className={style.button + " waves-effect waves-light btn"} onClick ={() => this.props.sortItems("ascending")}>Ascending</a>
            <a className={style.button + " waves-effect waves-light btn"} onClick ={() => this.props.sortItems("descending")}>Descending</a>
            </div>

        )
    }
}
```

Sort Component

```jsx
// sort the items according to the price
const sortItems = (order)=>{
  var sortArray= [];
  if (order == 'ascending')
  {

    sortArray= this.state.filteredItems.sort((a, b) => (a.price > b.price) ? 1 : -1)
    this.setState({filteredItems:sortArray});
  }
  else
  {

    sortArray= this.state.filteredItems.sort((a, b) => (a.price > b.price) ? -1 : 1)
    this.setState({filteredItems:sortArray});
  }

}
```

Home Component

Price Analysis-Graph.js Component

```
componentDidMount()
{

    const fetchPosts = async () => {
    await axios.get('/api/items').then(response=>
    this.setState({items:response.data},()=>console.log("products fetched", response.data)))

    this.setState({catgraph1:this.filter_array("khaadi")});
    this.setState({catgraph2:this.filter_array("outfitters")});
    this.setState({catgraph3:this.filter_array("breakout")});

    console.log(this.state.catgraph1);
    console.log(this.state.catgraph2);
    console.log(this.state.catgraph3);


};
fetchPosts();
}
filter_array(category)
{
    var array = this.state.items;
    let filt_array = array.filter(item => item.brand == category)
    filt_array =  filt_array.slice(0,30);
    let price=[];
    for (let i =0; i<filt_array.length;i++)
        price.push(filt_array[i].price)
    return price;
}
render(){
    return(
        <div>
            <Navbar />
            <div><h1 style={{textAlign:"center", color:"lightCoral"}}>Price Analysis </h1></div>
            <Chart data1={this.state.catgraph1} borderColor={"chocolate"} titleText={"Khaadi Price Analysis"} label={"Khaadi"}/>
            <Chart data1={this.state.catgraph2} borderColor={"lightSeaGreen"} titleText={"Outfitters Price Analysis"} label={"Outfitters"}/>
            <Chart data1={this.state.catgraph3} borderColor={"paleVioletRed"} titleText={"Breakout Price Analysis"} label={"Breakout"} />
            <Footer />
        </div>
```

Price Analysis- Chart Component

```
componentDidMount() {
setTimeout(() => {
    this.setState({
        chartData:{
            labels:['1','2','3','4','5','6','7','8','9','10','11',
            datasets:[
            {
                label:this.props.label,
                data: this.props.data1,
                fill:false,
                //lineTension:0,
                borderColor:this.props.borderColor,
                borderWidth:2,
                hoverBorderWidth:5,
            }]
        }


    })

}, 1000)
setTimeout(() => {
var array = this.props.data1;
var minval = Math.min.apply(null, array);
var maxval = Math.max.apply(null, this.props.data1);
var sum = this.props.data1.reduce((a, b) => a + b,0)
var avgval = (sum / this.props.data1.length).toFixed(2);
this.setState({min:minval, max:maxval, avg:avgval})
}, 1000)

}

// setting the default values of the props
static defaultProps = {
    titleText : "Brand Price Analysis",
    label:"brand",
    borderColor:'lightCoral'
}
```

```
setTimeout(() => {
var array = this.props.data1;
var minval = Math.min.apply(null, array);
var maxval = Math.max.apply(null, this.props.data1);
var sum = this.props.data1.reduce((a, b) => a + b,0)
var avgval = (sum / this.props.data1.length).toFixed(2);
this.setState({min:minval, max:maxval, avg:avgval})
}, 1000)

}

// setting the default values of the props
static defaultProps = {
    titleText : "Brand Price Analysis",
    label:"brand",
    borderColor:'lightCoral'
}

render(){
    console.log("chart "+this.props.data1)
            return(
        <div className='chart container'>
        <div className={style.chart}>
        <Line data={this.state.chartData} width={200} height={70} options={{
            title:{
                display:true,
                text:this.props.titleText,
                fontSize:25,
                padding:25,
                fontColor:'lightCoral',
            },
            legend:{
                display:true,
                position:'right'
            }
        }} />
         <ul className={style.box + " collection with-header"} style={{color:this
            <li className="collection-header" style={{backgroundColor:this.props.
            <li className="collection-item">Minimum Price: {this.state.min}</li>
            <li className="collection-item">Maximum Price: {this.state.max}</li>
            <li className="collection-item">Average Price: {this.state.avg}</li>
```

Feedback form
Contact component

```
import React, {Component} from 'react';
import Navbar from './Navbar.js';
import Form  from './Form.js';
import Footer from './Footer.js';
class Contact extends Component{
    constructor(props){
        super(props)
    }
    render(){
        return(
            <div>
                <Navbar />
                <div><h2 style={{textAlign:"center", color:"lightCoral"}}>Give Us Feedback!</h2></div>
                <Form />
                <Footer />
            </div>
        )
    }
}
export default Contact;
```

Feedback-Form
Form Component

```
changeHandler(event)
{
    var input = event.target;
    var name = input.name;
    var val = input.value;
    var message ="";

    if (name=="name")
    {
            if (!/^[a-zA-Z ]+$/.test(val))
            {
                this.messageN = "Invalid name";
            }
            else
            {
                this.messageN = "";
            }
    }
    else if (name=="phone")
    {
            if (!/^[+]*[(]{0,1}[0-9]{1,4}[)]{0,1}[-\s\./0-9]*$/g.test(val))
            {
                this.messageP = "Invalid Phone";

            }
            else
            {
                this.messageP = "";
            }

    }

    if (this.messageN == "" && this.messageP == "" )
    {
        this.setState({valid:true})
    }
    else
    {
        this.setState({valid:false})
    }
```

```
<form className={style.form +" col s12"} method="post" action="/api/form">
  <div className="row">
    <div className="input-field col s6">
      <i className="material-icons prefix">account_circle</i>
      <input id="icon_prefix" type="text" name="name" className="validate" onBlur={this.changeHandler} minLength="3" m
      <label for="icon_prefix">Name</label>
    </div>
    <div className="input-field col s6">
      <i className="material-icons prefix">phone</i>
      <input id="icon_telephone" type="tel" name="phone" className="validate" onBlur={this.changeHandler} required/>
      <label for="icon_telephone">Telephone</label>
    </div>
  </div>
  <div className="row">
    <div className="input-field col s12">
     <i className="material-icons prefix">contact_mail</i>
      <input id="email" type="email" name="email" className="validate" required/>
      <label for="email">Email</label>
    </div>
  </div>
  <div className="row">
     <div className ="col s12">
        <h4>Rate the website</h4>
          <p>
          <label>
          <input name="rate" type="radio" value ="Excellent" unchecked  />
          <span className={style.radio}>Excellent</span></label></p>
          <p>
          <label><input name="rate" type="radio" value ="Good" unchecked />
          <span className={style.radio}>Good</span></label></p>
          <p>
          <label><input name="rate" type="radio" value ="Average" unchecked  />
          <span className={style.radio}>Average</span></label></p>
          <p>
          <label><input name="rate" type="radio" value ="Poor" unchecked  />
          <span className={style.radio}>Poor</span></label></p>
```