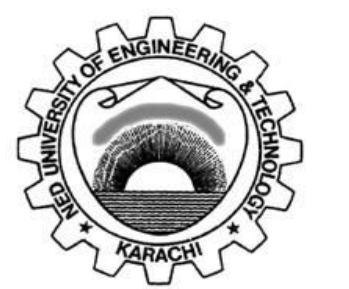
CS-222 - Database Management System

Complex Engineering Problem (CEP) - Report

Spring Semester 2021

HOMOFY

Ecommerce Website

### Group Members:

Muhammad Zain Ammad (CS-19079)

Syed Saad Ali (CS-19088)

Moiz Ali (CS-19087)

Areeb bin Azim (CS-19090

## Project Title

HOMOFY

## Abstract

HOMOFY is an Ecommerce Website for shopping. It will use database to store different products of an Online shopping like Clothing, Home appliances, Sports goods etc. It will store information about price, colors, material and other information regarding products. Moreover, it will also store information about admins and users.

### Task Distribution Chart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Zain | Moiz | Saad | Areeb |
| Database / schema / SQL | ✔ | ✔ |  | ✔ |
| React JS , Bootstrap | ✔ |  |  | ✔ |
| React Hooks (usestate , useeffect , usehistory , uselocation) | ✔ |  |  | ✔ |
| Data Entry | ✔ | ✔ | ✔ | ✔ |
| APIs (Axios , Fetch API) |  | ✔ |  | ✔ |
| Node JS |  | ✔ |  |  |
| Express JS |  | ✔ |  |  |
| Report writing |  | ✔ |  | ✔ |

### Folder Structure

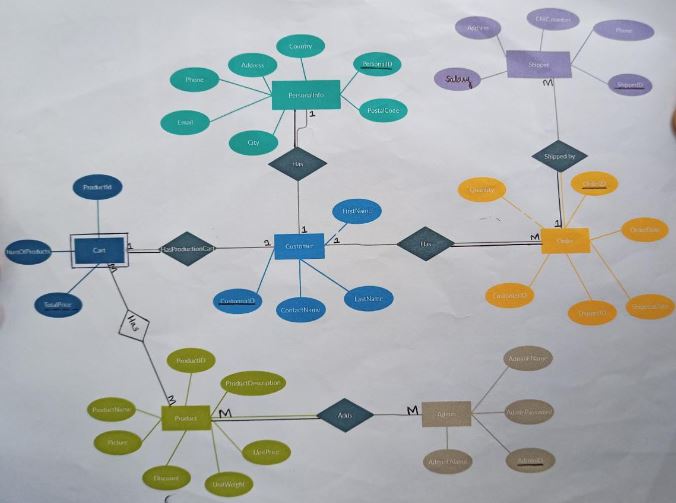
### Client:

Client folder deals with the front end of this project i.e React JS.

### Server:

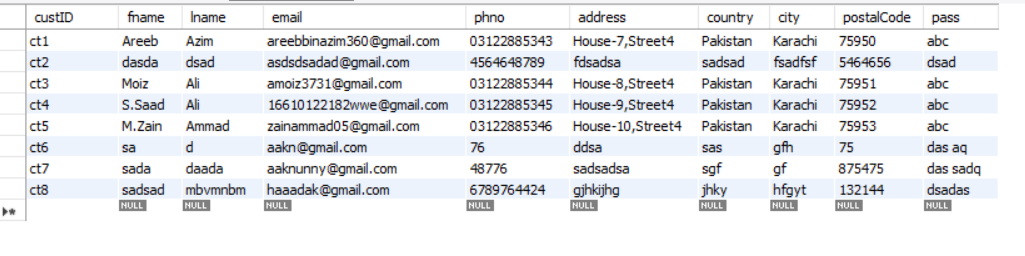
Server folder comprises of the back end of this project, having Node JS and Express JS.

**Relational Schema of HOMOFY Database:**

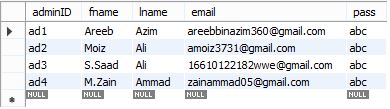


### Tables After Normalization:

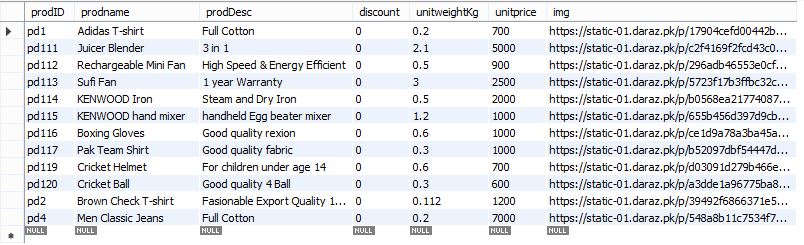
* **Customer:**



* **Admin:**

****

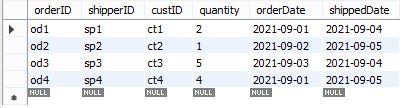
* **Products:**

****

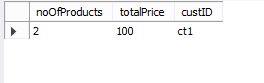
* **Shipper:**

****

* **Orders:**

****

* **Cart:**

****

**SQL Queries:**

* **Creating Table:**

-- CREATE TABLE admin(

-- adminID varchar(20) NOT NULL UNIQUE,

-- fname varchar(50) NOT NULL,

-- lname varchar(50),

-- email varchar(60) NOT NULL UNIQUE,

-- pass varchar(30) NOT NULL

-- );

-- CREATE TABLE cart(

-- cartID varchar(20) NOT NULL UNIQUE,

-- noOfProducts int NOT NULL,

-- totalPrice int NOT NULL

-- );

-- CREATE TABLE shipper(

-- shipperID varchar(20) NOT NULL UNIQUE,

-- cnic varchar(30) NOT NULL UNIQUE,

-- phno varchar(30) NOT NULL UNIQUE,

-- address varchar(60) NOT NULL,

-- salary int

-- );

-- CREATE TABLE customer(

-- custID varchar(20) NOT NULL UNIQUE,

-- fname varchar(50) NOT NULL,

-- lname varchar(50),

-- email varchar(60) NOT NULL UNIQUE,

-- phno varchar(30) NOT NULL UNIQUE,

-- address varchar(60) NOT NULL,

-- country varchar(30) NOT NULL DEFAULT 'Pakistan',

-- city varchar (30) NOT NULL,

-- postalCode int NOT NULL

-- );

-- CREATE TABLE product(

-- prodID varchar(20) NOT NULL UNIQUE,

-- prodname varchar(50) NOT NULL,

-- prodDesc varchar(100) NOT NULL,

-- discount int NOT NULL DEFAULT 0,

-- unitweightKg float NOT NULL,

-- unitprice int NOT NULL

-- );

-- CREATE TABLE orders(

-- orderID varchar(20) NOT NULL UNIQUE,

-- shipperID varchar(20) NOT NULL UNIQUE,

-- custID varchar(20) NOT NULL UNIQUE,

-- quantity int NOT NULL,

-- orderDate DATE NOT NULL,

-- shippedDate DATE

-- );

* **Inserting data into tables:**

-- INSERT INTO admin (adminID,fname,lname,email,pass)

-- VALUES

-- ('ad1','Areeb','Azim','areebbinazim360@gmail.com','abc'),

-- ('ad2','Moiz','Ali','amoiz3731@gmail.com','abc'),

-- ('ad3','S.Saad','Ali','16610122182wwe@gmail.com','abc'),

-- ('ad4','M.Zain','Ammad','zainammad05@gmail.com','abc');

-- INSERT INTO cart (cartID,noOfProducts,totalPrice)

-- VALUES

-- ('cart1',2,20),

-- ('cart2',4,50),

-- ('cart3',3,70),

-- ('cart4',2,20);

-- INSERT INTO customer (custID,fname,lname,email,phno,address,country,city,postalCode)

-- VALUES

-- ('ct1','Areeb','Azim','areebbinazim360@gmail.com','03122885343','House-7,Street4','Pakistan','Karachi',75950),

-- ('ct2','Moiz','Ali','amoiz3731@gmail.com','03122885344','House-8,Street4','Pakistan','Karachi',75951),

-- ('ct3','S.Saad','Ali','16610122182wwe@gmail.com','03122885345','House-9,Street4','Pakistan','Karachi',75952),

-- ('ct4','M.Zain','Ammad','zainammad05@gmail.com','03122885346','House-10,Street4','Pakistan','Karachi',75953);

-- INSERT INTO shipper (shipperID,cnic,phno,address,salary)

-- VALUES

-- ('sp1','123','03123456567','House-7,Street8',15000),

-- ('sp2','456','03123456568','House-8,Street7',16000),

-- ('sp3','789','03123456569','House-9,Street6',25000),

-- ('sp4','0321','03123456560','House-10,Street5',18000);

-- INSERT INTO product (prodID,prodname,prodDesc,discount,unitweightKg,unitprice)

-- VALUES

-- ('pd1','Adidas T-shirt','Full Cotton',0,0.2,700),

-- ('pd2','Denim Jeans Pant','Wash Faded quality',0,0.5,1100),

-- ('pd3','Rado Mens Watch','Silver Plated',0,0.22,2000),

-- ('pd4','Nike Mens Shoes','Very Light',0,0.54,3000);

-- INSERT INTO orders (orderID,shipperID,custID,quantity,orderDate,shippedDate)

-- VALUES

-- ('od1','sp1','ct1',2,'2021-09-01','2021-09-04'),

-- ('od2','sp2','ct2',1,'2021-09-02','2021-09-05'),

-- ('od3','sp3','ct3',5,'2021-09-03','2021-09-04'),

-- ('od4','sp4','ct4',4,'2021-09-01','2021-09-05');

-- ALTER TABLE orders

-- ADD FOREIGN KEY(shipperID) REFERENCES shipper(shipperID),

-- ADD FOREIGN KEY(custID) REFERENCES customer(custID);

### Backend

The backend of this project is implemented in python’s Flask framework. Flask listens for requests from the front end, performs some logic like data validation, database querying, etc and sometimes (but not always) responds with some data. We have python functions for all routes of the website. A route is a link that can be visited like <website\_name>/phones/favourites.

Flask involves writing “templates” with “jinja” language for the front end which are HTML pages with space for different information which can be filled on run time, for example the individual phone page’s HTML has a table with space for specifications of phone, space for reviews, etc.

Brief explanation of all routes is given as follows:

#### /

The default route leads to the homepage.

#### /signup and /signin

If they receive a GET request, they return the respective HTML page. If they receive a POST request, they validate the information submitted via the form on frontend and check against the user table in the database. If successful, the user is redirected to the homepage, else an error message is flashed onto the screen.

#### /signout

Logs users out, by forgetting them from the cookie session and redirects to the home page.

#### /brands

Accepts GET requests only. Loads information of all brands, and returns the brand page with that information filled.

#### /brands/add

If a GET request is received, return the add brand page. If POST request is received, as via clicking submit on add brand page, then validates the form data and checks if brand exists or not. If successful, the brand is added and redirects to /brands, else the page is reloaded with an error message flashed.

#### /brands/<brand\_name> (eg: /brands/Xiaomi)

Returns the HTML document of the brand in the url by first fetching data related to it from the database.

#### /brands/<brand\_name>/edit (eg: /brands/Xiaomi/edit)

If a GET request is received, then querying the database for the brand information and renders the form with that data for the admin to edit. If a POST request is received, validates form data and if successful, saves new data to the database for that brand.

#### /brands/<brand\_name>/delete

If a GET request is received, it renders a warning page. If a POST request is received and the yes button was clicked then that brand is deleted from the database alongside all of its phones.

#### /phones

Accepts GET requests only. Loads information of all phones from the database and returns the Phones page with that information filled. If GET parameters related to filters like ram, brand, etc are received too then the database is queried according to the constraints of those filters.

#### /phones/favourites

Similar to /phones but only shows the favourite phones of the current user.

#### /phone/add

Similar to /brands/add but for phones instead.

#### /phones/<brand\_phone\_id> (eg: /phones/Xiaomi\_Mi-11-Lite\_21)

Querying the database for all information of the phone and loads pages of that phone.

#### /phones/<brand\_phone\_id>/edit

Similar to /brands/<brand\_name>/edit but for a phone.

#### /phones/<brand\_phone\_id>/delete

Similar to /brands/<brand\_name>/delete but for a phone.

#### /phones/<brand\_phone\_id>/add-review

If a GET request, then simply renders the page with a form for review. Else if POST request, then checks received review data, validates, and if successful, adds review for that phone to database and then redirects to the respective phone’s page

#### /news

Queries database for all news and renders a page with all of them displayed.

#### /news/add

If GET requests, then renders a page with a form for adding a news. Else if POST request then validates data and if successful, adds news to database and redirects to /news.

#### /news/<heading\_id>(eg: http://127.0.0.1:5000/news/Xiaomi\_Mi\_Pad\_5\_appears\_in\_official\_teaser\_with\_keyboard\_accessory-1)

Queries database for the news and renders the full news page for that news.

#### /news/<heading\_id>/delete

Deletes the news and redirects to /news

#### /delete-review

It's an API route and it doesn't return any page. Just deletes the review which was requested to be deleted.

#### /favouriteify\_phone

It's an API route and it doesn't return any page. Just makes the requested phone a favourite or non-favourite (like / unlike) for the user logged in.

### 

### Frontend

The frontend involves HTML for page structure (quite a bit of the frontend HTML is generated by Flask from backend); CSS for styling; JavaScript and JQuery for some frontend logic and talking with the backend. JQuery in particular was used for AJAX calls for tasks that require sending additional requests to the backend without the need of reloading the entire page like favoriting a phone or deleting a review.

Pictures (and explanation of some) of the front end are shown as follows:

## 

## 

**Details**

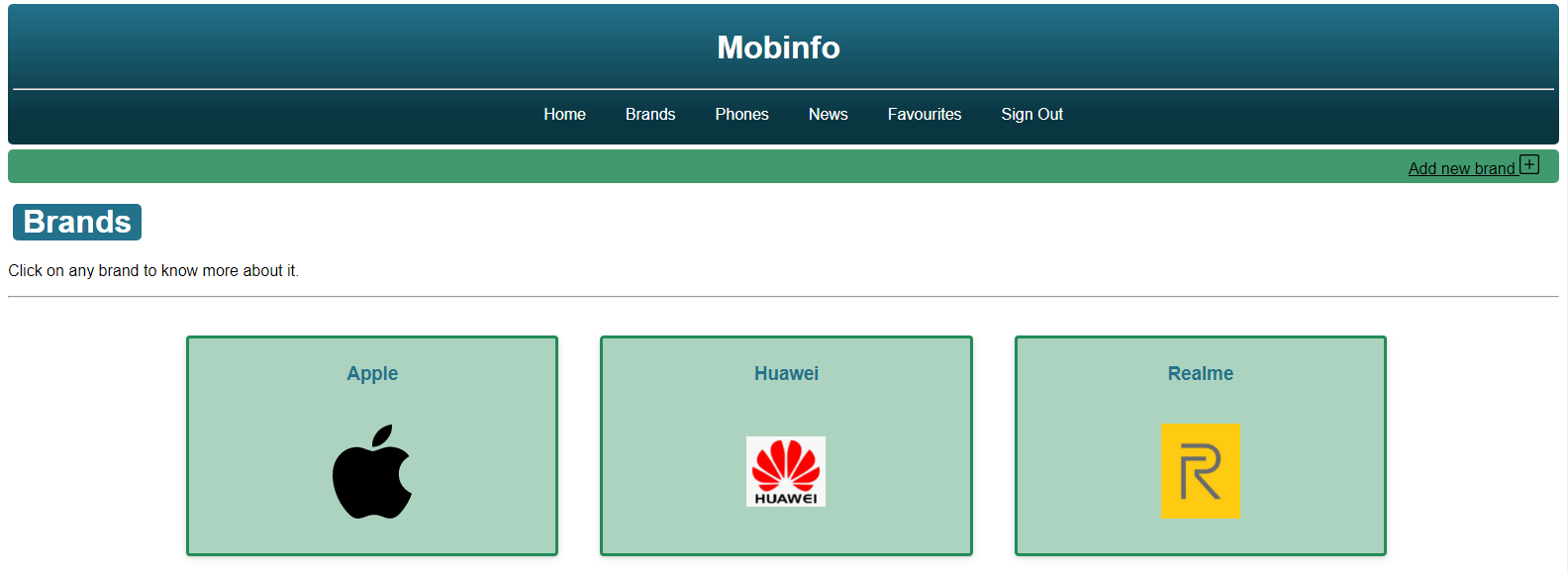
### Home

This is the first page you will see, when you open our website. It intends to put a great impression on users with its Mobile wallpapers carousel; beautiful header and navbar; and status of user login on footer. It also includes some of our top picked phones. It is the main page from where you can go to different parts of the website as you desire, like if you want to see the Phones, you can click “Phones” on the navbar.

### 

### Brands

This page shows all the brands whose phones are present on the site. Any brand can be clicked to view more information about it (Individual brand page, explained below)

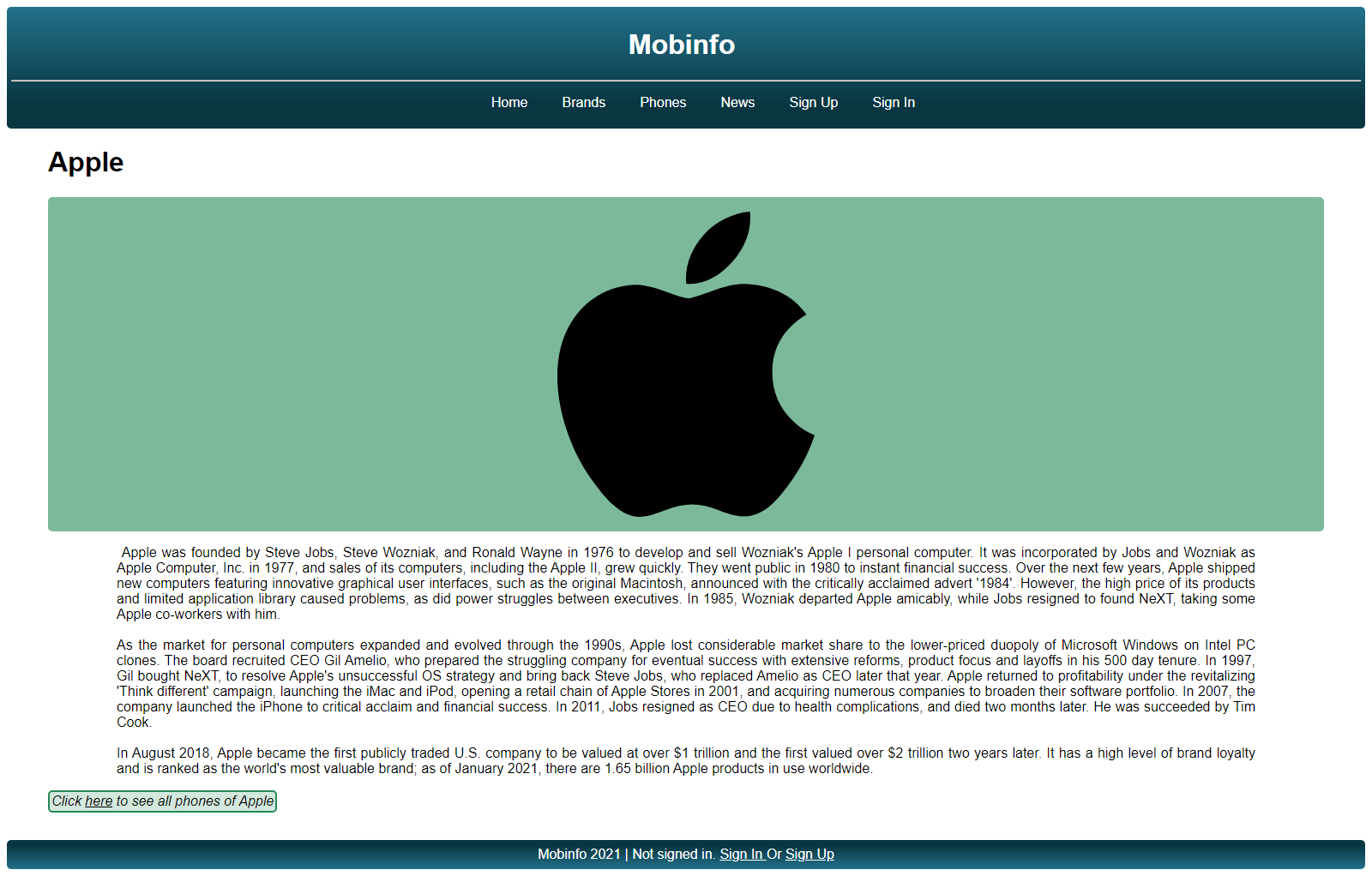


If logged in as admin, you can also see the “Add new brand” button.



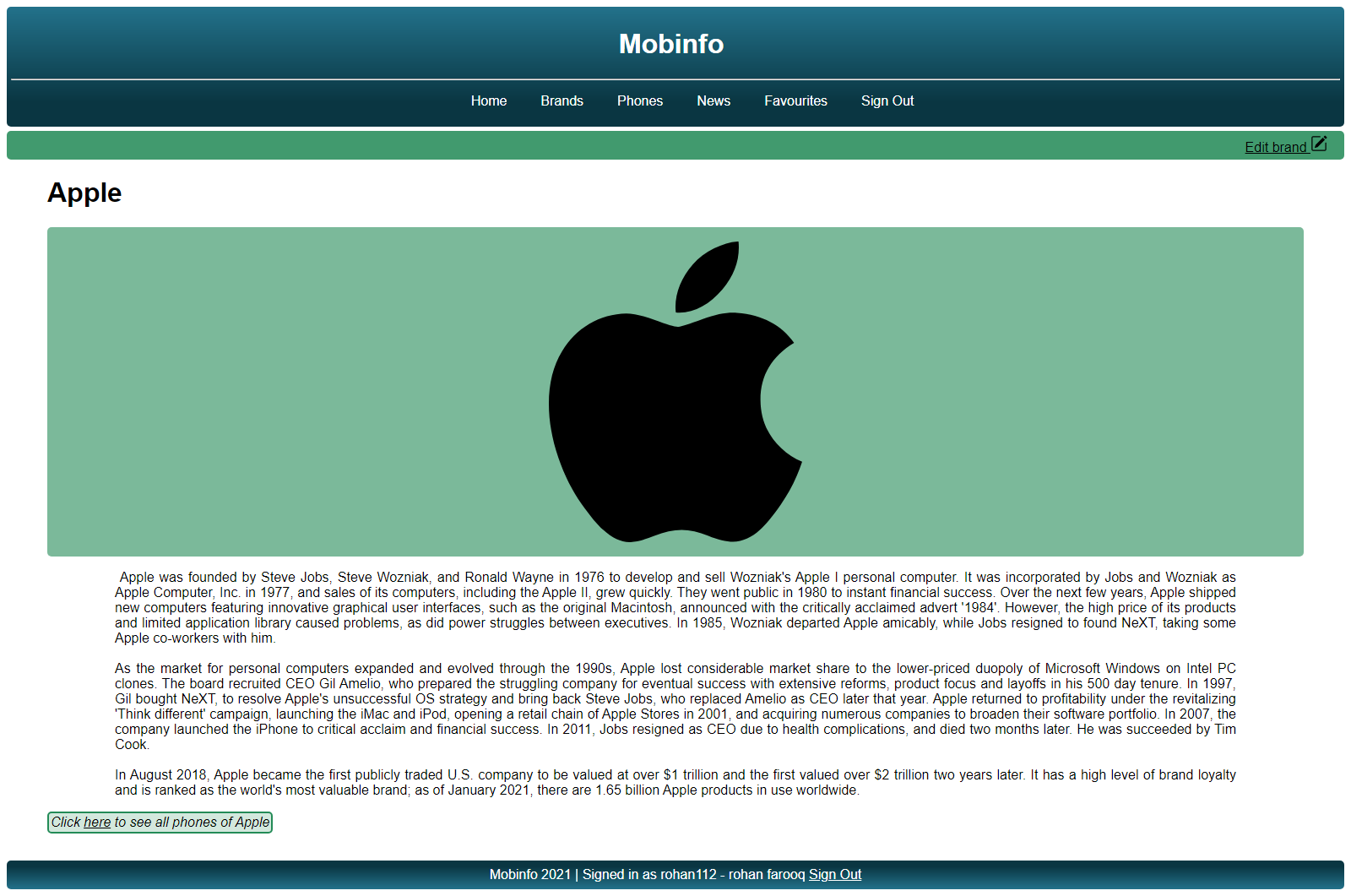
Add Brand

A form is given with fields for details of the new brand to be added.

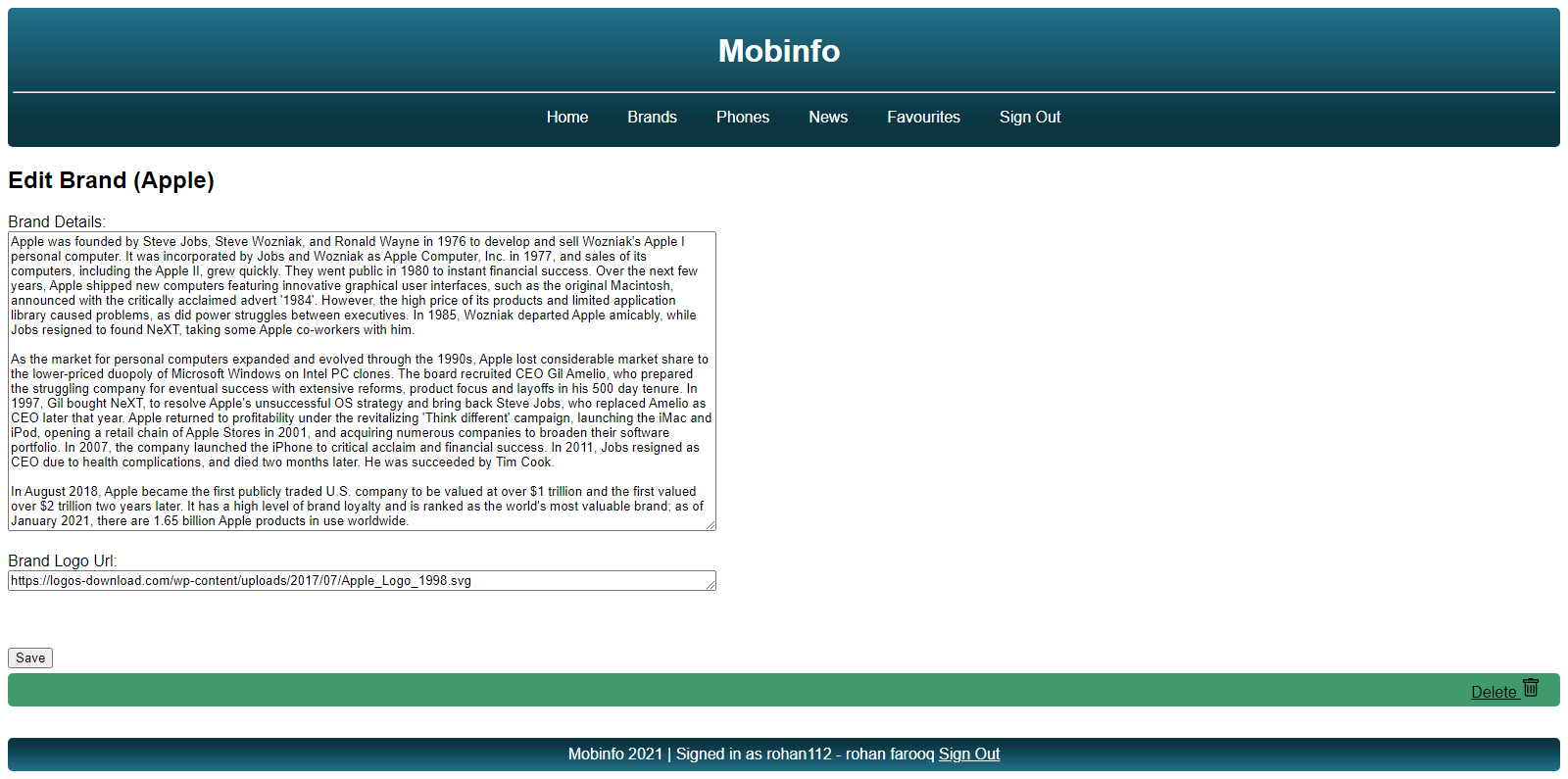
After submitting the form, depending on whether the form was properly filled and brand was added or brand was not added, a contextual feedback message is flashed on to the screen.

Individual Brand

On clicking any brand on the Brands page, this page opens and it has the logo of the brand, some information about it, and a link that can be clicked to view phones of that brand



If logged in as admin, the edit brand button is also visible.

Edit Individual Brand

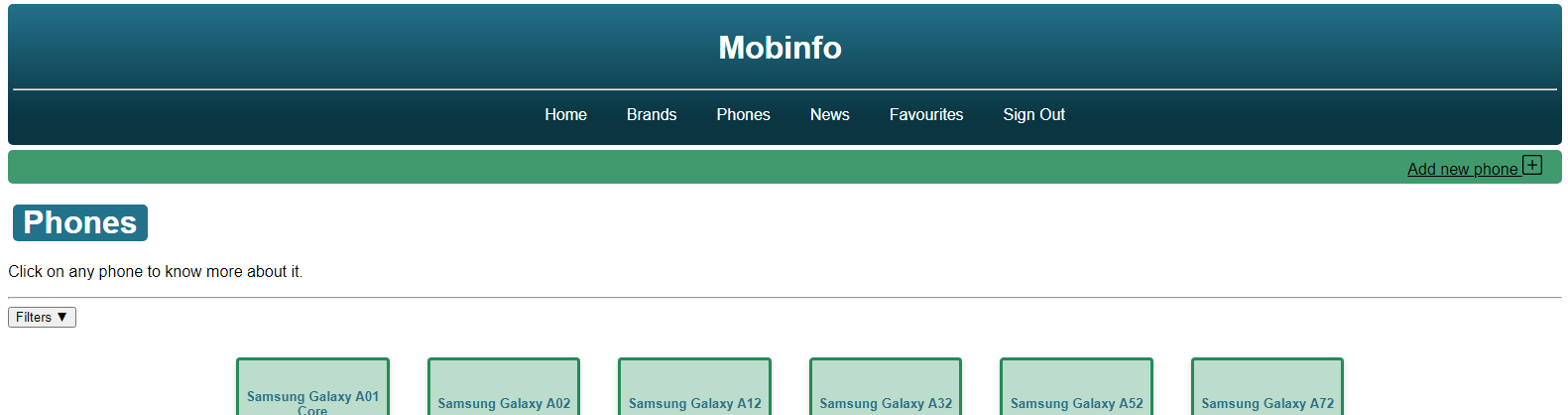
A form is given with fields for details of the brand being edited pre filled for the admin to change.

### 

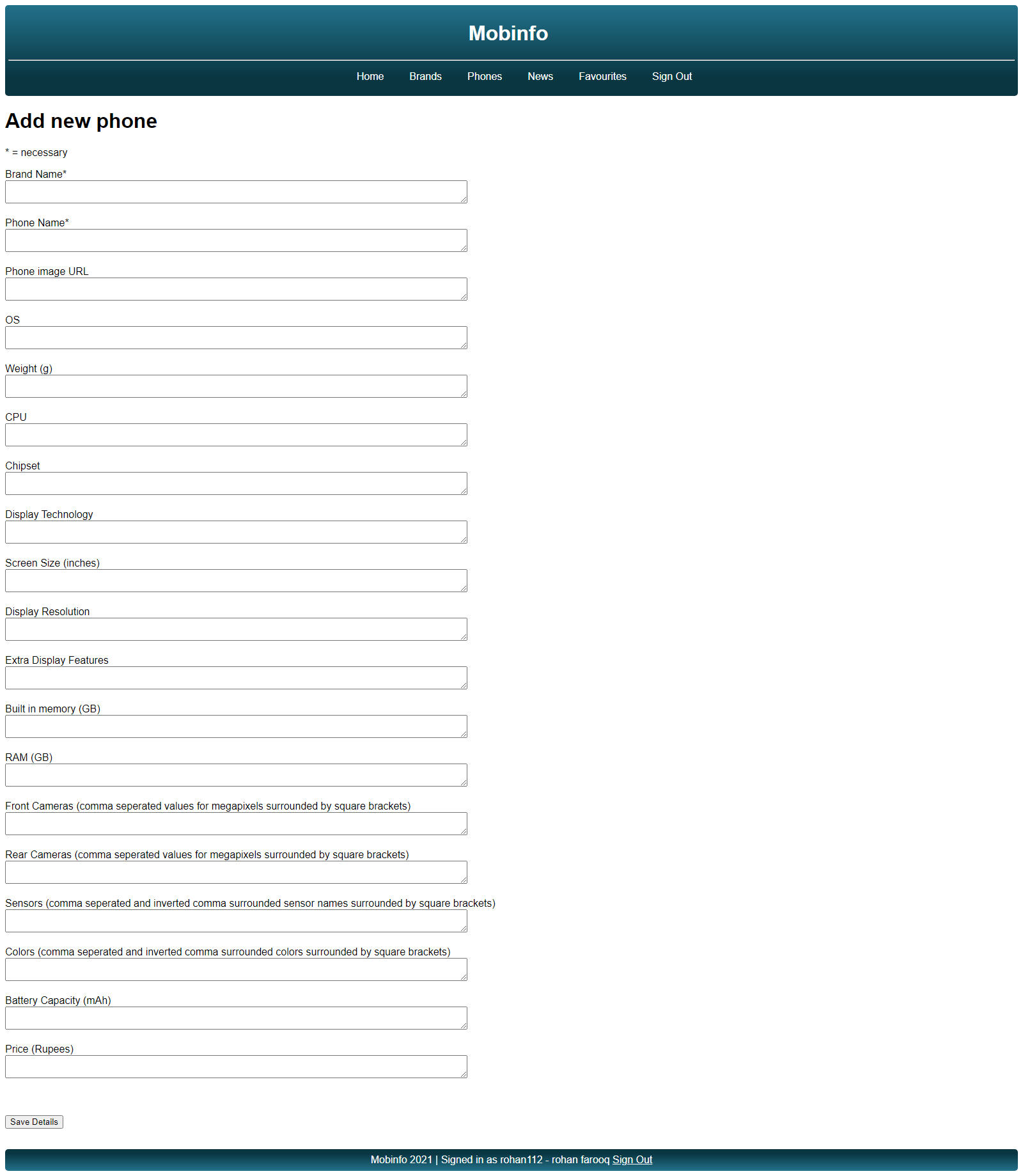
### 

### Phones

This page has all brand phones in which users can see the different brand phones with its name and a picture.

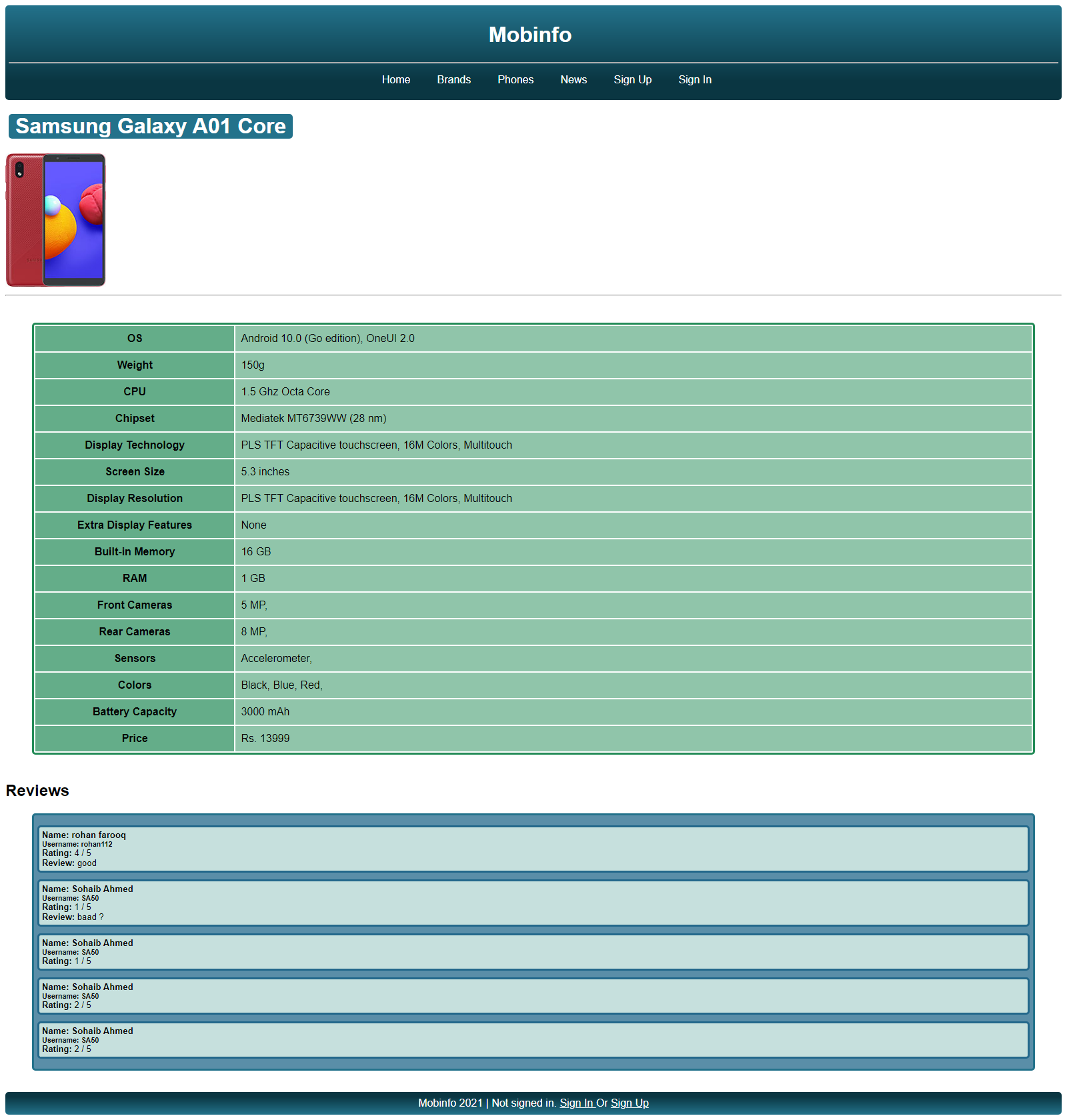


An add new phone button is visible on top if logged in as admin



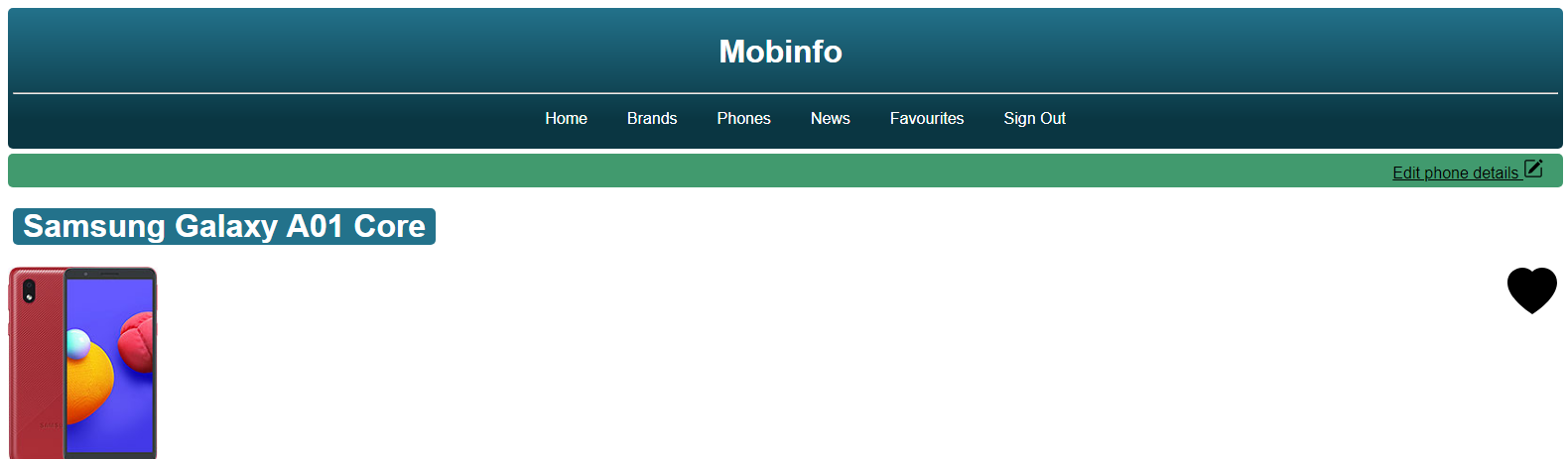
Add Phones

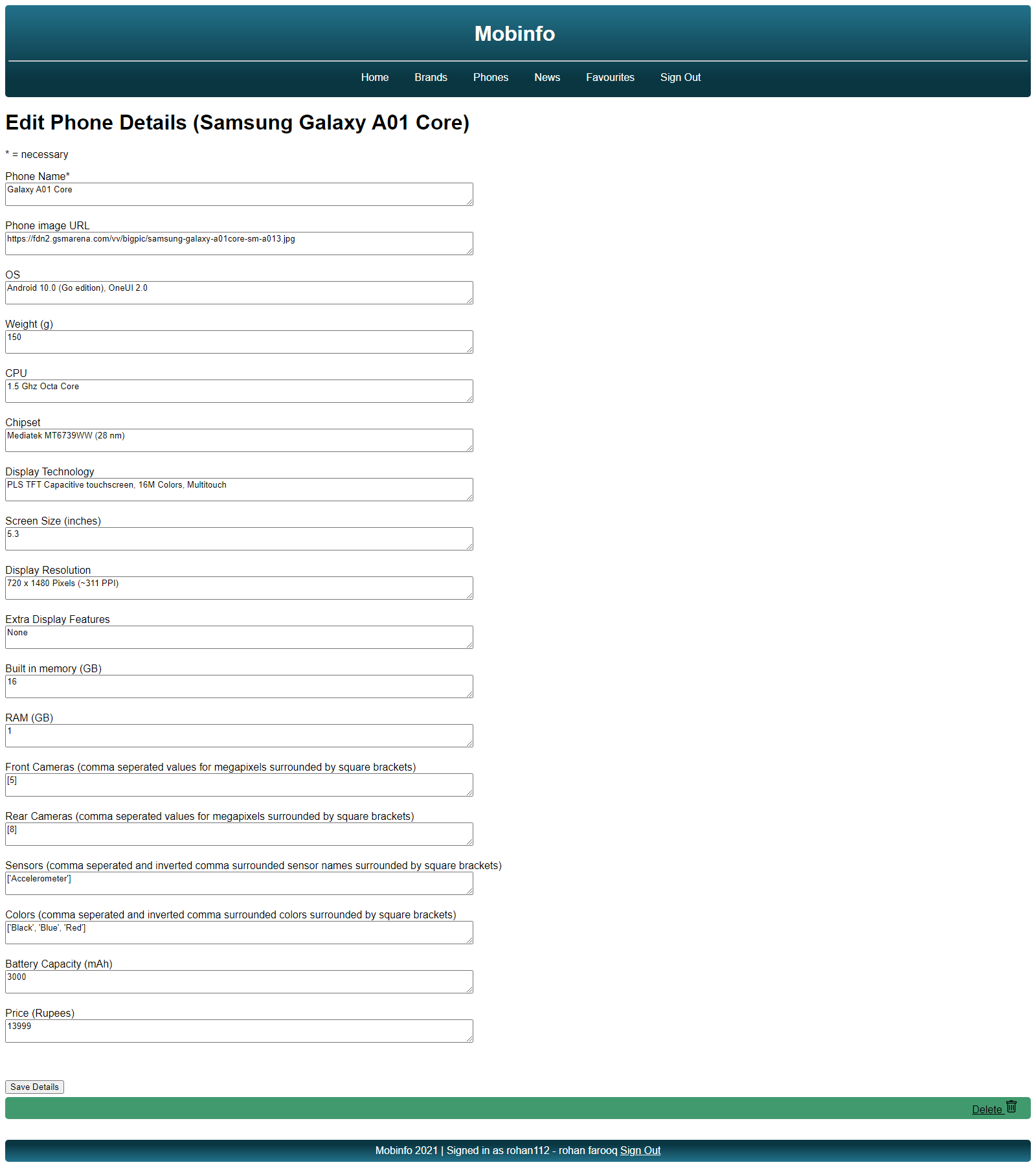
As an admin, you can add a new phone by filling up the form on the add new phone page (part of which is shown on the right).



Individual Phone

Clicking any phone on the Phones page opens up this page which has details of the phone followed by reviews and option to add review if logged in. The page also has a heart button to favourite a phone if logged in.

If logged in as admin, the edit phone button is visible too.

Edit Individual Phone

A form is given with fields for details of the phone being edited pre-filled for the admin to change.

### News

This page contains news about phones, brands, and different technologies.

If logged in, the add news button is visible too

Add News

As an admin, you can add new news by filling up the Heading, Image URL and Text of News in which Heading is compulsory to fill.

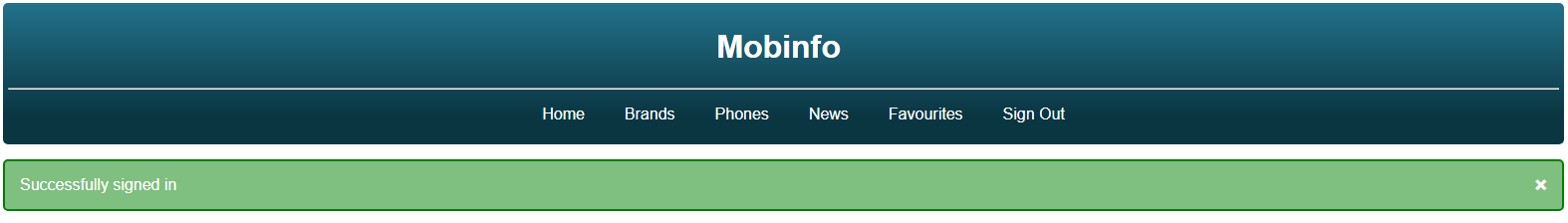
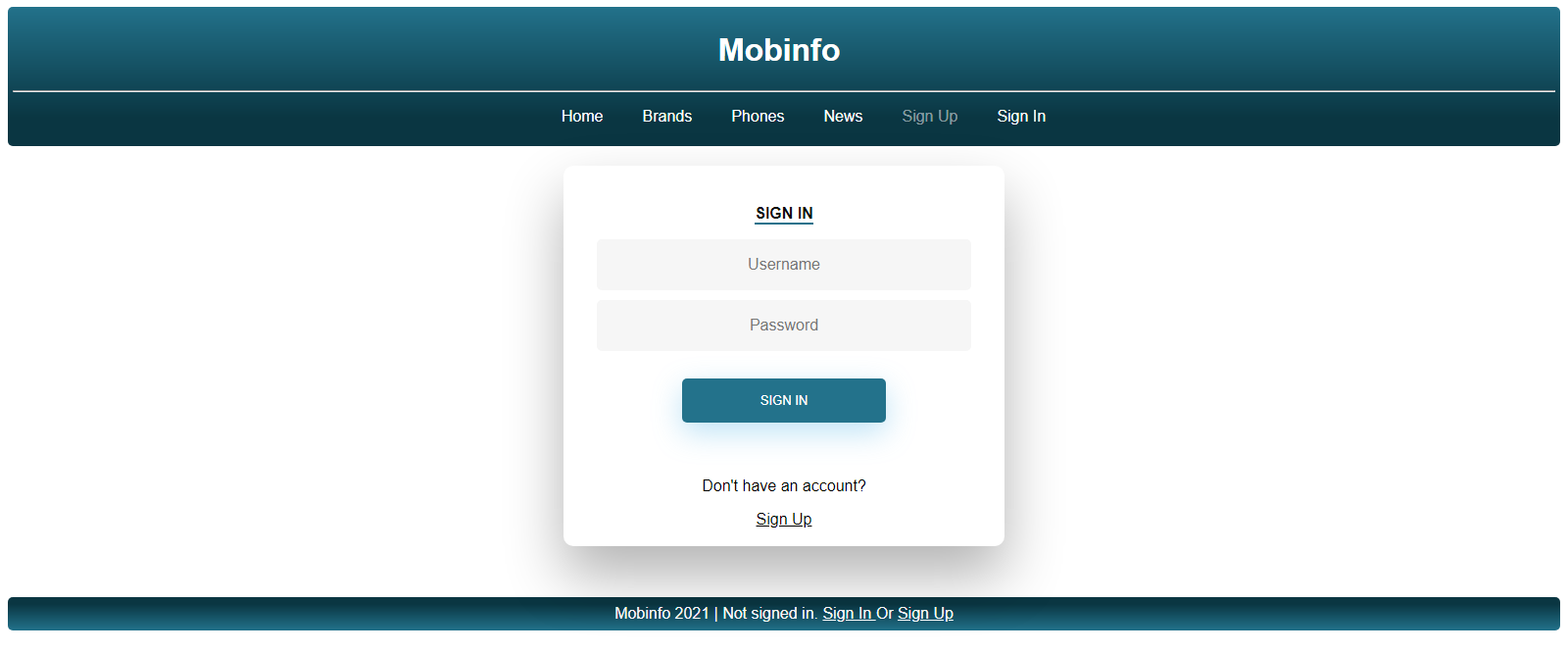


Individual News

When you click the one of each news item then a new page will appear which has the complete information about news.

### Favourites

If logged in, you can visit the favourites page to view the phones you have liked

**Sign In**

This is the Sign In page of our website. For logging in, you must have your account on our website and if you have an account, you just have to use your username and password to log in.

If you enter both username and password correctly then the contextual feedback message will also show.

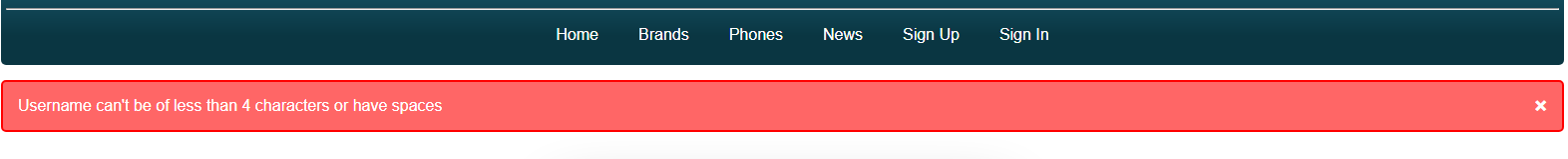
If you enter the wrong username or the warning will also show

### Sign Up

In this page, you can make your account of our website. For this, you just have to enter Your First Name, Last Name, Unique username of length greater than 4, password of length greater than 6.

If you enter all information correctly, then the contextual feedback message will also show.

And if not then warning according to your mistake will show.



### 

