



CSE471: System Analysis and Design Project Report

Project Title: Mobile Benchmarking System

Group No: 11, CSE471 Lab Section: 01, Fall 2024	
ID	Name
21201080	Md. Abrar Rahman Shafin
23341115	Yasin Rahman
21301083	Torsha Bashar

Submission Date: 7 May, 2025

Table of Contents

1. System Request	3
Business need:	3
Business requirements:	3
Business value:	4
Special issues or constraints:	4
2. Functional Requirements	4
3. Technology (Framework, Languages)	5
4. Backend Development	5
5. User Interface Design	12
6. Frontend Development	17
7. User Manual	24
8. Performance and Network Analysis	31
9. Github Repo [Public] Link	33
10. Link of Deployed Project	33
11. Individual Contribution	34
12. References	35

1. System Request

Business need: The Mobile Benchmarking System project has been initiated to assist customers in making informed smartphone purchasing decisions by providing a centralized platform for comparing phone specifications, user reviews, performance benchmarks, and price updates. This will help increase user engagement, boost affiliate sales through third-party vendor redirects, and establish TechCompare Ltd. as a go-to source for mobile technology evaluations.

Business requirements: The system must enable users and administrators to interact with mobile phone data and provide tools for intelligent comparison and decision-making. Core functionality should include:

- **User Authentication** (registration and login via email and password).
- **Admin Panel** for managing phone entries (add/edit/delete with full specs).
- **Phone Specification Display** including processor, RAM, storage, display, OS, battery, camera, etc.
- **Multiple image view with auto slideshow**, and similar phone suggestions.
- **News & Updates Section** for daily coverage of smartphone launches.
- **Phone Purchase Options** with redirect links to platforms like Amazon and Best Buy.
- **User Reviews & Q&A** including upvotes, filters, and feedback sorting.
- **Feature-Specific Ratings** (camera, battery, display, performance) with average scores and a sidebar of top 5 highest-rated phones.
- **Wishlist Functionality** and price drop notifications.
- **Search & Filtering** based on specs, price range, and manufacturers.
- **Phone Comparison Tool** allowing 2-device side-by-side analysis with calculated benchmark scores and a summary of strengths.
- **Trending & Value Phones Sidebar** displaying top 5 trending and value-for-money smartphones.

Business value:

The Mobile Benchmarking System is expected to drive value through increased user traffic, improved affiliate sales, and enhanced customer satisfaction by offering a reliable, feature-rich comparison platform. The combination of technical benchmarking, user reviews, and purchase guidance positions the system to be a leader in the mobile tech evaluation niche.

Conservative estimates of tangible value include the following:

- \$500,000 in affiliate revenue from phone purchase redirection.
- \$300,000 increase in ad revenue from higher site engagement.
- \$120,000 estimated yearly value from brand partnerships and sponsored placements.
- Enhanced brand equity and long-term customer loyalty from delivering trusted, useful comparison features.

Special issues or constraints:

- The platform must support high performance and responsiveness due to heavy media content and real-time filtering.
- Data integrity and accuracy are critical as users rely on this for decision-making.
- Real-time or scheduled updates from third-party sources (pricing/news) must be reliable.
- A competitive launch is essential to capture market share before similar tools enter the market.

2. Functional Requirements

- 1. User Authentication**
 - Users can register/login using email and password.
- 2. Phone Database Management (Admin Panel)**
 - Admin can add new phones with all specifications.
 - Admins can edit/update phone details.
 - Admins can delete phone entries.
- 3. Phone Specification Display**
 - Each phone has a dedicated page displaying:
 - Processor, RAM, Storage, Battery, Display, OS, Camera, etc.
 - Multiple Image + Auto Slideshow
 - Similar phone suggestions
- 4. News & Updates Section**
 - Daily smartphone updates covering new launches.
- 5. Phone Purchase Option**
 - Users can view purchasing options (e.g., Amazon, Best Buy).
 - Redirect users to third-party sellers
- 6. User Reviews**
 - Users can leave reviews
 - Users can ask questions about specific devices.
 - Users can upvote helpful reviews and filter based on highest ratings or most relevant feedback.
- 7. User-Scored Feature Ratings**

- Users can rate individual core features of a phone, like camera, battery, display, and performance.
- An average score will be displayed for each feature based on user ratings.
- Top 5 highest rated phones will be shown on the side bar.

8. Save Phones to wishlist

- Users can save their favorite phones to their wishlist to get price update notification.

9. Search & Filtering

- Search by key specifications such as RAM, storage, battery life, display type, and camera quality.
- Sort results by lowest to highest and highest to lowest price.
- Users can set a minimum and maximum budget to find phones within their price range.
- Find phones by selecting specific manufacturers like Apple, Samsung, Xiaomi, etc.

10. Comparison

- Users can compare 2 phones side by side to analyze their performance differences.

11. Benchmark Feature

- Provides calculated score between two phones after comparison.
- Shows which phone is the winner.
- An analysis section that shows strengths of both phones.

12. Price Update Notification

- Users will get notifications of the wishlisted phones if price changes

13. Top Phones Sidebar

- Top 5 Trending Phones (based on user rating)
- Top 5 Value for Money Phones (based on calculated score and given price)

3. Technology (Framework, Languages)

Language: JavaScript

Techstack: MERN (MongoDB, Express, React, Node)

4. Backend Development

View Phone Details:

Users can view prices, features such as camera, display, os, processor etc by clicking on any phone that is available in the phone database.

```
router.get('/:phoneId', getPhoneDetails);

exports.getPhoneDetails = async (req, res) => {
  const { phoneId } = req.params;

  try {
    const phone = await Phone.findById(phoneId);
    if (!phone) return res.status(404).send('Phone not found');
    res.send(phone);
  } catch (err) {
    res.status(500).send(err.message);
  }
}
```

```

        const phone = await Phone.findById(phoneId).lean({ virtuals: true
    }) ;

    if (!phone) {
        return res.status(404).json({ error: 'Phone not found' });
    }

    res.status(200).json(phone);
} catch (err) {
    console.error(err);
    res.status(500).json({ error: 'Server error' });
}
};

}

```

The screenshot shows a Postman interface with the following details:

- Method:** GET
- URL:** <http://localhost:1080/api/phones/680808c642d0811957f1b4b4>
- Headers:** (8)
- Body:** (JSON) - Preview shows the following JSON response:

```

1  {
2      "_id": "680808c642d0811957f1b4b4",
3      "name": "Samsung Galaxy S23 Ultra",
4      "brand": "Samsung",
5      "price": 1999,
6      "features": {
7          "camera": {
8              "main": 200,
9              "ultraWide": 12,
10             "front": 12,
11             "videoRecording": "8K"
12         },

```

Delete Phone:

Admin can delete any phone.

```

router.delete('/:id', deletePhone);

exports.deletePhone = async (req, res) => {
    try {
        const deletedPhone = await Phone.findByIdAndDelete(req.params.id);
        if (!deletedPhone) {
            return res.status(404).json({ error: 'Phone not found' });
        }
        res.json({ message: 'Phone deleted successfully' });
    } catch (error) {
        res.status(500).json({ error: error.message });
    }
}

```

```
};
```

The screenshot shows a REST client interface. At the top, there's a header bar with 'DELETE' and a URL 'http://localhost:1080/api/phones/67f3ad26e54eab6f17455412'. Below the header are tabs for 'Params', 'Authorization', 'Headers (8)', 'Body' (which is selected), 'Scripts', and 'Settings'. Under 'Body', there are options for 'none', 'form-data', 'x-www-form-urlencoded', 'raw' (which is selected), 'binary', 'GraphQL', and 'JSON'. To the right of these options are 'Cookies' and 'Beautify' buttons. The main body area contains a code editor with three lines of JSON: '1 {', '2 | "message": "Phone deleted successfully"', and '3 }'. Below the code editor, there are tabs for 'Body', 'Cookies', 'Headers (8)', and 'Test Results'. The 'Test Results' tab shows a green success icon. To the right of the results, there are buttons for 'Copy', 'Close', 'Search', and 'Link'. At the bottom right of the interface, there's a small preview window showing the JSON response.

Rating:

Users can rate individual phone features.

```
router.post('/add', addRating)

exports.addRating = async (req, res) => {
  try {
    const rating = await Rating.create(req.body);
    res.status(201).json(rating);
  } catch (err) {
    res.status(400).json({ error: err.message });
  }
};
```

The screenshot shows a Postman interface with a POST request to `http://localhost:1080/api/ratings/add`. The request body is a JSON object:

```

1  {
2   "phone": "67f271abb184462b2e4b16f1",
3   "user": "john_doe",
4   "ratings": {
5     "camera": 4,
6     "battery": 5,
7     "display": 4,
8     "processor": 5
9   },
10  "comment": "Great phone overall!"
11 }

```

The response is a 201 Created status with the following JSON data:

```

1  {
2   "phone": "67f271abb184462b2e4b16f1",
3   "user": "john_doe",
4   "ratings": {
5     "camera": 4,
6     "battery": 5,
7     "display": 4,
8     "processor": 5
9   },
10  "comment": "Great phone overall!",
11  "_id": "67f3afbbe54eab6f1745541d",
12  "createdAt": "2025-04-07T10:58:03.689Z",
13  "updatedAt": "2025-04-07T10:58:03.689Z",
14  "__v": 0
15 }

```

Users can see the average rating of all rated features.

```

router.get('/:phoneId/average', getAverageRatings);

exports.getAverageRatings = async (req, res) => {

  try {

    const phoneId = req.params.phoneId;

    const result = await Rating.aggregate([
      { $match: { phone: new mongoose.Types.ObjectId(phoneId) } },
      {
        $group: {
          _id: '$phone',
          avgCamera: { $avg: '$ratings.camera' },
          avgBattery: { $avg: '$ratings.battery' },
          avgDisplay: { $avg: '$ratings.display' },
          avgProcessor: { $avg: '$ratings.processor' },
          count: { $sum: 1 }
        }
      }
    ]);

    res.json(result);
  } catch (err) {
    console.error(err);
    res.status(500).json({ error: 'Internal Server Error' });
  }
}

function getAverageRatings(req, res) {
  const phoneId = req.params.phoneId;
}

```

```

        }

    }

] );

if (!result.length) {

    return res.status(404).json({ error: 'No ratings found for this phone.' });
}

res.status(200).json(result[0]);

} catch (err) {

    res.status(500).json({ error: err.message });
}

};

}

```

GET <http://localhost:1080/api/ratings/67f271abb184462b2e4b16f1/average> Send

Params Authorization Headers (8) **Body** Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```

1  {
2      "phone": "67f271abb184462b2e4b16f1",
3      "user": "john_doe",
4      "ratings": {
5          "camera": 4,
6          "battery": 5,
7          "display": 4,
8          "processor": 5
9      },
10     "comment": "Great phone overall!"
11 }

```

Body Cookies Headers (8) Test Results

200 OK • 47 ms • 385 B • [Raw](#) [JSON](#) [Copy](#) [Link](#)

{ } JSON Preview Visualize

```

1  {
2      "_id": "67f271abb184462b2e4b16f1",
3      "avgCamera": 3.25,
4      "avgBattery": 4.25,
5      "avgDisplay": 3.25,
6      "avgProcessor": 4.75,
7      "count": 4
8 }

```

Users can search by name, sort, filter phones.

```
router.get('/', getPhones);
exports.getPhones = async (req, res) => {
  try {
    const {
      search,
      brand,
      sort,
      minPrice,
      maxPrice,
      minRam,
      minStorage,
      minBattery,
      minRefreshRate,
      minCamera,
      os,
      network,
      // sortByValueScore,
    } = req.query;

    const query = {};

    // Search by name or processor
    if (search) {
      query.$or = [
        { name: { $regex: search, $options: 'i' } },
        { 'features.processor.name': { $regex: search, $options: 'i' } }
      ];
    }

    // Filter by brand
    if (brand) {
      query.brand = { $regex: brand, $options: 'i' };
    }

    // Filter by price range
    if (minPrice || maxPrice) {
      query.price = {};
      if (minPrice) query.price.$gte = Number(minPrice);
      if (maxPrice) query.price.$lte = Number(maxPrice);
    }

    // Filter by minimum RAM
    if (minRam) {
      query['features.memory.ram'] = { $gte: Number(minRam) };
    }

    // Filter by minimum Storage
    if (minStorage) {
      query['features.memory.storage'] = { $gte: Number(minStorage) };
    }

    // Filter by battery capacity
    if (minBattery) {
```

```

        query['features.battery.capacity'] = { $gte: Number(minBattery) }
    };
}

// Filter by display refresh rate
if (minRefreshRate) {
    query['features.display.refreshRate'] = { $gte: Number(minRefreshRate) };
}

// Filter by camera quality (main camera)
if (minCamera) {
    query['features.camera.main'] = { $gte: Number(minCamera) };
}

// Filter by OS
if (os) {
    query['features.os'] = { $regex: os, $options: 'i' };
}

// Filter by network type (like 5G, 4G)
if (network) {
    const networks = Array.isArray(network) ? network : [network];
    query.$or = networks.map((net) => ({
        'features.network': { $regex: new RegExp(net, 'i') }
    }));
}

let phonesQuery = Phone.find(query);

// Sorting
if (sort) {
    phonesQuery = phonesQuery.sort({ price: sort === 'asc' ? 1 : -1 });
}

const phones = await phonesQuery.lean({ virtuals: true });

res.status(200).json(phones);
} catch (err) {
    res.status(500).json({ error: err.message });
}
};

}

```

HTTP <http://localhost:1080/api/phones?brand=Apple&minRam=4&minStorage=64&minCamera=12&sort=asc>

GET <http://localhost:1080/api/phones?brand=Apple&minRam=4&minStorage=64&minCamera=12&sort=asc>

Save Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Body Cookies Headers (8) Test Results ⏱

200 OK • 51 ms • 1.65 KB • ⌂ ⏷ ⏸ ⏹ ⏺ ⏻

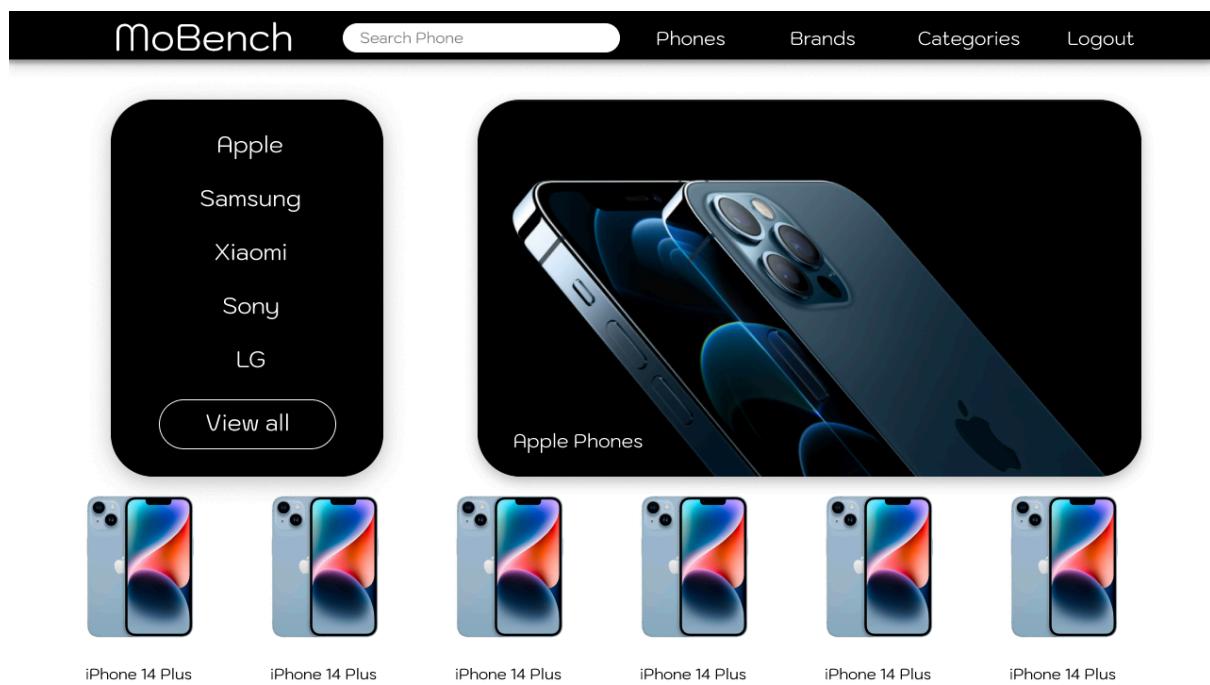
{ } JSON ▾ Preview Visualize

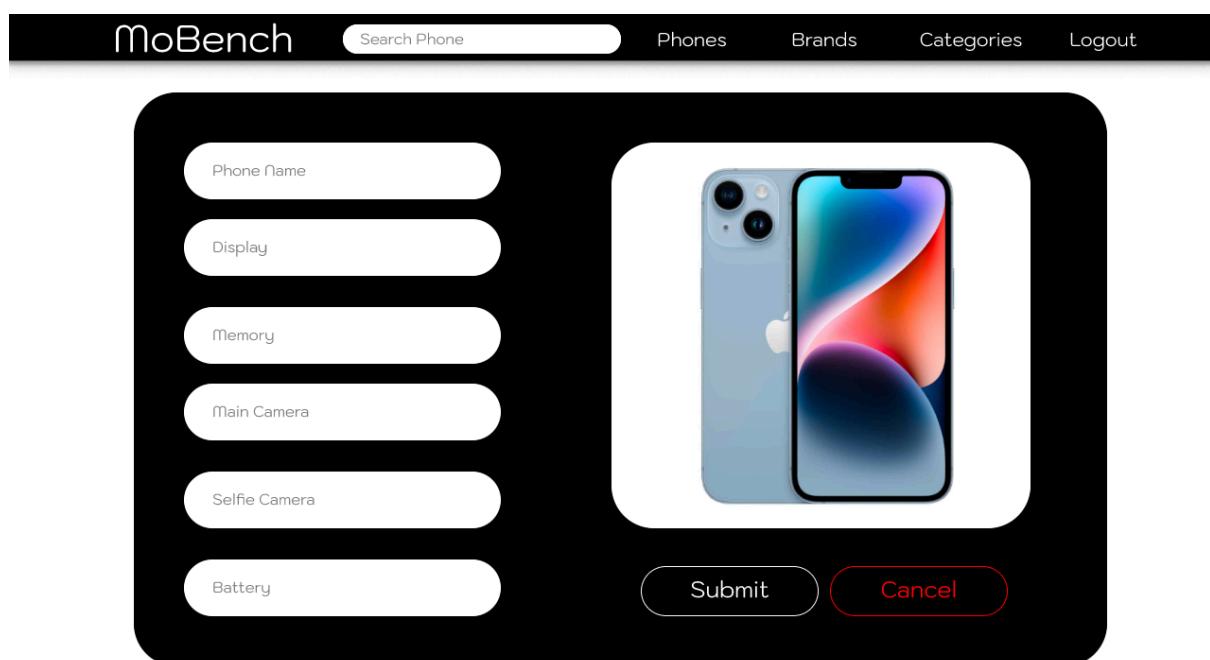
```

1  [
2    {
3      "_id": "68080a2142d0811957f1b4d4",
4      "name": "iPhone 15 Pro Max",
5      "brand": "Apple",
6      "price": 1199,
7      "features": {
8        "camera": {
9          "main": 48,
10         "ultraWide": 12,
11         "front": 12,
12         "videoRecording": "4K60"
13       },
14       "battery": {
15         "capacity": 4422,
16         "chargingSpeed": 27,
}

```

5. User Interface Design







Apple iPhone 14 Plus

Released 2022, October 07

203g, 78mm thickness

iOS 16, up to iOS 18.3.2

128GB/256GB/512GB storage, no card slot

Display

6.7 inches, 109.8 cm² (~87.4% screen-to-body ratio)



Platform

Apple GPU (5-core graphics)



Main Camera

12 MP, f/2.4, 13mm, 120° (ultrawide)



Selfie Camera

12 MP, f/1.9, 23mm (wide)



[Home](#) / [Cell Phones and Accessories](#) / [Cell Phones](#) / [Oppo A5](#)



Oppo A5 Pro 8/128GB

Status: IN STOCK

Choose The best deal

- Tk 25,000
- Tk 24,500
- Tk 28,000

Similar Products



Oppo A3

Tk-22,000 -25,000



Oppo A3

Tk-22,000 -25,000



Oppo A3

Tk-22,000 -25,000



Oppo A3

Tk-22,000 -25,000

[Specification](#) [Description](#) [Review](#) [Questions\(3\)](#)

Description

Oppo A5

The Oppo A5 Pro is a powerful smartphone designed to offer an exceptional experience for users who demand both performance and style. Featuring a stunning 6.67-inch LCD display with a 90Hz refresh rate and a peak brightness of 1000 nits, this phone ensures vivid, smooth visuals whether you're browsing, gaming, or streaming content. Powered by the Snapdragon 6s chipset and 8GB of RAM, the Oppo A5 Pro delivers fast performance that handles multiple apps and demanding tasks with ease. Its 128GB of internal storage provides ample space for your apps, photos, and videos, while the microSD card slot offers further expansion options for those who need extra storage.

Phone Kinbo

☰ Menu

What can we help you find today?

Search

Account

0

Home / Cell Phones and Accessories / Cell Phones / Oppo A5



Oppo A5 Pro 8/128GB
Status: IN STOCK
Choose The best deal
Tk 25,000
Tk 24,500
Tk 28,000

Specification Description Review Questions(3)

Specification

RAM	3/4GB
INTERNAL STORAGE	64 GB
SIM	Dual Sim
USB	microUSB 2.0
Operating System	Android 12 or 13 (Go edition), MIUI

Similar Products

Oppo A3
Tk-22,000 -25,000

Oppo A3
Tk-22,000 -25,000

Oppo A3
Tk-22,000 -25,000

Oppo A3
Tk-22,000 -25,000

Phone Kinbo

☰ Menu

What can we help you find today?

Search

Account

0

User Review and Ratings- Oppo A5

★★★★★

Brilliant smartphone

This smartphone is a very awesome and nice battery life this smartphone is for who plays games till 6-8hrs

Sami Hossain
11/10/25

Is this review helpful?
 (3) (1)

★★★★★

Opp a5 excellent phone

I have this mobile since 1.5 year and i realise that it is very good mobile.its storage is a problem but by using SD card we can fix it all over good....

Fabiba Anber Ahmed
11/10/25

Is this review helpful?
 (3) (1)

★★★☆☆

oppo a5 not a good phone

not a good phone equal as realme 2 everything is same except 32 gb memory and price is high go for real me 2 and 27 sept realising wait for real me 2 pro in same price for abetter phone

Sahaf Sayeed
11/10/25

Is this review helpful?
 (3) (1)

★★★☆☆

Buy other smartphones

Oppo A5 has a lower display resolution and a slower processor, poor camera quality and laggy while playing games. You can buy other better models with better features and higher performance.

Mosaovir Ahmed Khan
11/10/25

Is this review helpful?
 (3) (1)

Frequently Asked Questions

1. Can I use online storage with my Oppo A5?

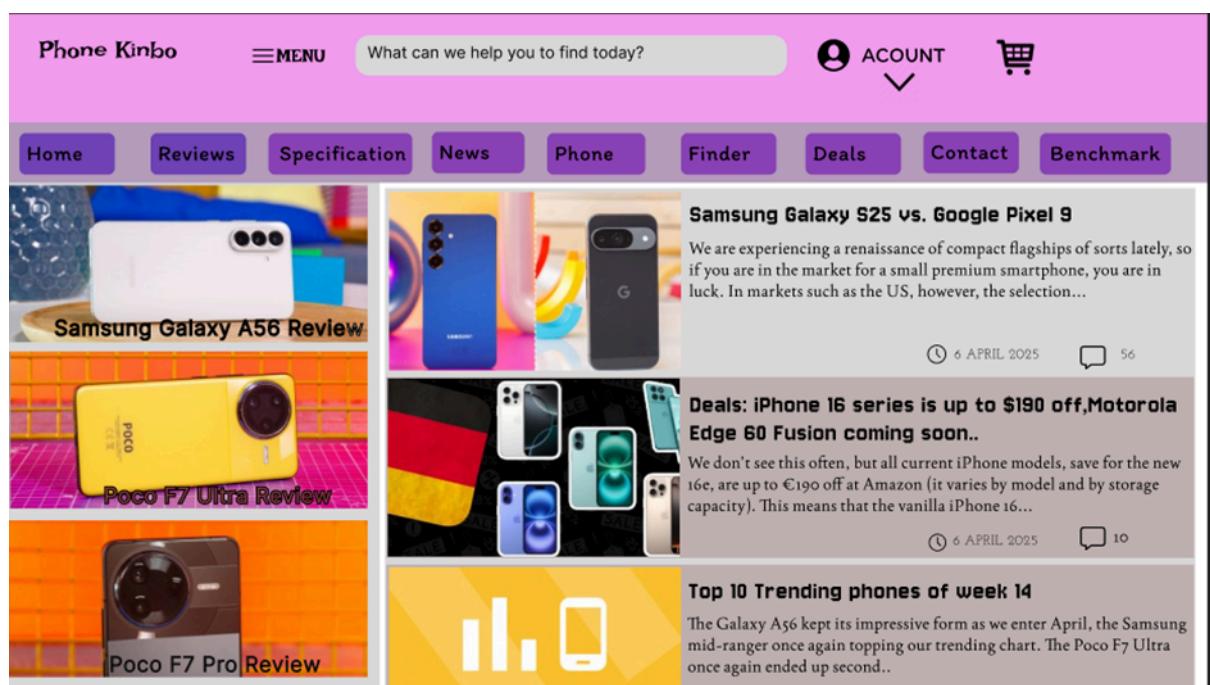
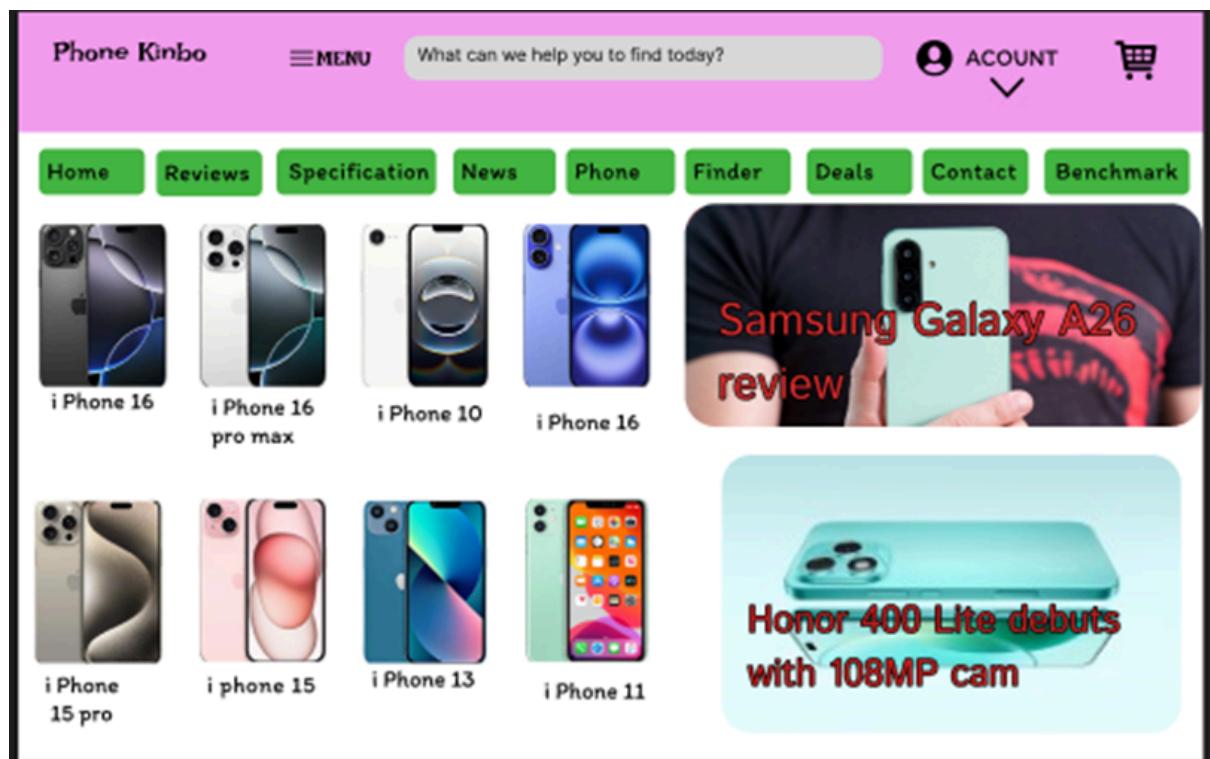
Besides the internal storage of your device, you can also use Cloud Storage. Here you can upload your photos and videos, so they no longer have to be on your phone. They remain available on your phone via the Cloud, but do not take up any more space on your phone. In this way, your phone will stay fast and your storage will become much less full. Examples of Cloud storage are iCloud, Dropbox, Google Drive, and Microsoft OneDrive.

2. How do I disable my talkback on Oppo A5?

It is possible that TalkBack is turned on and because of this your device does not respond to your touches. This is a feature of Android, made for the blind and visually impaired. In this mode, the phone converts the text on the screen into speech. You can disable this feature from the settings menu, under Accessibility.

3. How do I disable my talkback on Oppo A5?

It is possible that TalkBack is turned on and because of this your device does not respond to your touches. This is a feature of Android, made for the blind and visually impaired.



Figma Links:

[21201080_11_Assignment2 – Figma](#)
[23341115_11_Assignment2 - Figma](#)
[21301083_11_Assignment2 - Figma](#)

6. Frontend Development

This is the frontend login component implemented using Firebase. It accepts the user's email and password as input and includes a login button to initiate the authentication process. The login component also includes a registration link that redirects users to the registration form if they don't have an account.

```
import React, { useState } from 'react';
import './style.css';
import { Link, useNavigate } from 'react-router-dom';
import { auth } from '../firebase';
import { signInWithEmailAndPassword } from 'firebase/auth';

const Login = () => {
    const [email, setEmail] = useState('');
    const [password, setPassword] = useState('');
    const navigate = useNavigate();

    const handleSubmit = async (e) => {
        e.preventDefault();
        try {
            await signInWithEmailAndPassword(auth, email, password);
            console.log("Logged in Successfully");
            navigate('/home');
        } catch (err) {
            console.log(err);
            alert("Login failed. Please check your email and password.");
        }
    };

    return (
        <div className="signup-container">
            <form className="signup-form" onSubmit={handleSubmit}>
                <h2>Login</h2>
                <label htmlFor="email">
                    Email:
                    <input type="text" onChange={(e) => setEmail(e.target.value)} />
                </label>
                <label htmlFor="password">
                    Password:
                    <input type="password" onChange={(e) => setPassword(e.target.value)} />
                </label>
            </form>
        </div>
    );
}

export default Login;
```

```

        </label>
        <button type='submit'> Login</button>
        <p>Don't have an account? <Link
to="/signup">Register</Link></p>
    </form>
</div>
);
};

export default Login;

```

This is the frontend code for the navigation bar, which includes the Logo, News, Wishlist, Notifications, Benchmark, FAQs, and a Logout button.

```

import React, {useState} from 'react';
import { Link, useNavigate } from 'react-router-dom';
import './Navbar.css';
import { getAuth, signOut } from 'firebase/auth';

const Navbar = ({ onOpenPhoneFinder }) => {
    const navigate = useNavigate();
    const [count] = useState(0)

    const handleLogout = () => {
        const auth = getAuth();
        signOut(auth)
            .then(() => navigate('/login'))
            .catch((error) => console.error('Logout failed:', error));
    };

    return (
        <nav className="navbar">
            <div className="navbar-logo">
                <Link to="/home">Mobilytics</Link>
            </div>
            <div className="navbar-links">
                <Link to="/news">News</Link>
                <Link to="/wishlist">Wishlist</Link>
                <Link to="/notifications">Notifications {count > 0 && <span
className="badge">{count}</span>}</Link>
                <Link to="/benchmark">Benchmark</Link>
                <Link to="/faq">FAQs</Link>
            </div>
        </nav>
    );
}

export default Navbar;

```

```

        <button onClick={handleLogout}
      className="logout-button">Logout</button>
    </div>
  </nav>
);
};

export default Navbar;

```

This is the frontend code for the **Notifications** component. It displays user-specific notifications fetched using the Firebase user ID, allows marking them as read, and updates the UI accordingly.

```

import React, { useEffect, useState } from 'react'
import axios from 'axios'
import { getAuth } from 'firebase/auth'

export default function Notifications() {
  const [notes, setNotes] = useState([])

  useEffect(() => {
    const user = getAuth().currentUser
    if (!user) return
    axios
      .get('http://localhost:1080/api/notifications', { params: { user: user.uid } })
      .then(res => setNotes(res.data))
      .catch(console.error)
  }, [])

  const markRead = async (id) => {
    await
    axios.put(`http://localhost:1080/api/notifications/${id}/read`)
    setNotes(n => n.map(x => x._id === id ? { ...x, read: true } : x))
  }

  return (
    <div className="notifications">
      <h2>Notifications</h2>
      {notes.length ? (
        <ul>
          {notes.map(n => (
            <li key={n._id} className={n.read ? 'read' : 'unread'}>
              {n.message}
            </li>
          ))}
        </ul>
      ) : (
        <p>No notifications available.</p>
      )}
    </div>
  )
}

export default Notifications;

```

```

        { !n.read && <button onClick={() => markRead(n._id)}>Mark
Read</button>
      </li>
    ) )
  </ul>
) : (
<p>No notifications.</p>
)
</div>
)
}

```

This is the frontend code for the **TrendingPhones** component. It fetches and displays a list of top-rated phones based on user ratings. Each phone entry includes the name, average rating, and total rating count, and links to the phone's detail page. A loading message is shown while the data is being fetched.

```

import React, { useEffect, useState } from 'react';
import axios from 'axios';
import { Link } from 'react-router-dom';
import './TrendingPhones.css'; // 👍 import your CSS file

const TrendingPhones = () => {
  const [trendingPhones, setTrendingPhones] = useState([]);
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    const fetchTrending = async () => {
      try {
        const res = await
axios.get('http://localhost:1080/api/phones/trending');
        setTrendingPhones(res.data);
      } catch (error) {
        console.error('Failed to fetch trending phones:', error);
      } finally {
        setLoading(false);
      }
    };
    fetchTrending();
  }, []);

  if (loading) return <div className="text-center p-4">Loading trending
phones...</div>;
}

```

```

return (
  <div className="trending-container">
    <h2 className="trending-header">🔥 Top Rated</h2>
    <ul className="trending-list">
      {trendingPhones.map((item) => (
        <li key={item.phone._id} className="trending-item">
          <Link to={`/phone/${item.phone._id}}`}>
            <span className="phone-name">{item.phone.name}</span>
            <span className="rating">★
              {item.averageRating.toFixed(1)} ({item.count})</span>
            </Link>
          </li>
        ))}
      </ul>
    </div>
  );
};

export default TrendingPhones;

```

This is the frontend code for the **ValueForMoney** component. It fetches and displays a list of top value-for-money phones based on a calculated value score. Each phone entry shows the name and score, and links to the phone's detail page. A loading message is shown while data is being retrieved.

```

import React, { useEffect, useState } from 'react';
import axios from 'axios';
import { Link } from 'react-router-dom';
import './TrendingPhones.css';

const ValueForMoney = () => {
  const [phones, setPhones] = useState([]);
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    const fetchValuePhones = async () => {
      try {
        const res = await
      axios.get('http://localhost:1080/api/phones/top-value');
        setPhones(res.data);
      } catch (error) {

```

```

        console.error('Failed to fetch value-for-money phones:', error);
    } finally {
        setLoading(false);
    }
};

fetchValuePhones();
}, []);
```

if (loading) return <div className="text-center p-4">Loading value-for-money phones...</div>;

```

return (
<div className="trending-container">
    <h2 className="trending-header">⚡ Value for Money</h2>
    <ul className="trending-list">
        {phones.map((phone, index) => (
            <li key={phone._id} className="trending-item">
                <Link to={`/phone/${phone._id}`} className="trending-link">
                    <span className="phone-name">{phone.name}</span>
                    <span className="rating">📈 Score:<br/>{phone.valueScore}</span>
                </Link>
            </li>
        )));
    </ul>
</div>
);
};

export default ValueForMoney;
```

This frontend code displays the user's wishlist, fetched using their Firebase UID. It shows phone name, image, and price, with links to detailed pages. Users can also remove phones from the wishlist, which updates both the backend and the frontend view.

```

import React, { useEffect, useState } from 'react'
import axios from 'axios'
import { getAuth } from 'firebase/auth'
import { Link } from 'react-router-dom'
import './Wishlist.css'
```

```

export default function Wishlist() {
  const [items, setItems] = useState([])

  useEffect(() => {
    const user = getAuth().currentUser
    if (!user) return
    axios
      .get('http://localhost:1080/api/wishlist', { params: { user: user.uid } })
      .then(res => setItems(res.data))
      .catch(console.error)
  }, [])
}

const handleRemove = async (id) => {
  if (!window.confirm('Remove this item from wishlist?')) return
  try {
    await axios.delete(`http://localhost:1080/api/wishlist/${id}`)
    setItems(prev => prev.filter(i => i._id !== id))
  } catch (err) {
    console.error('Remove failed', err)
    alert('Failed to remove')
  }
}

return (
  <div className="wishlist-container">
    <h2>Your Wishlist</h2>
    {items.length > 0 ? (
      <div className="wishlist-grid">
        {items
          .filter(w => w.phone)
          .map(w => (
            <div key={w._id} className="wishlist-card">
              <Link to={`/phone/${w.phone._id}`}>
                <img
                  src={w.phone.imageUrls?.[0] ||
'https://via.placeholder.com/150'}
                  alt={w.phone.name}
                  className="wishlist-thumbnail"
                />
              <div className="wishlist-info">
                <h3>{w.phone.name}</h3>
                <p className="wishlist-price">${w.phone.price}</p>
              
```

```

        </div>
      </Link>
      <button
        className="wishlist-remove"
        onClick={() => handleRemove(w._id)}
      >
        Remove
      </button>
    </div>
  ))}

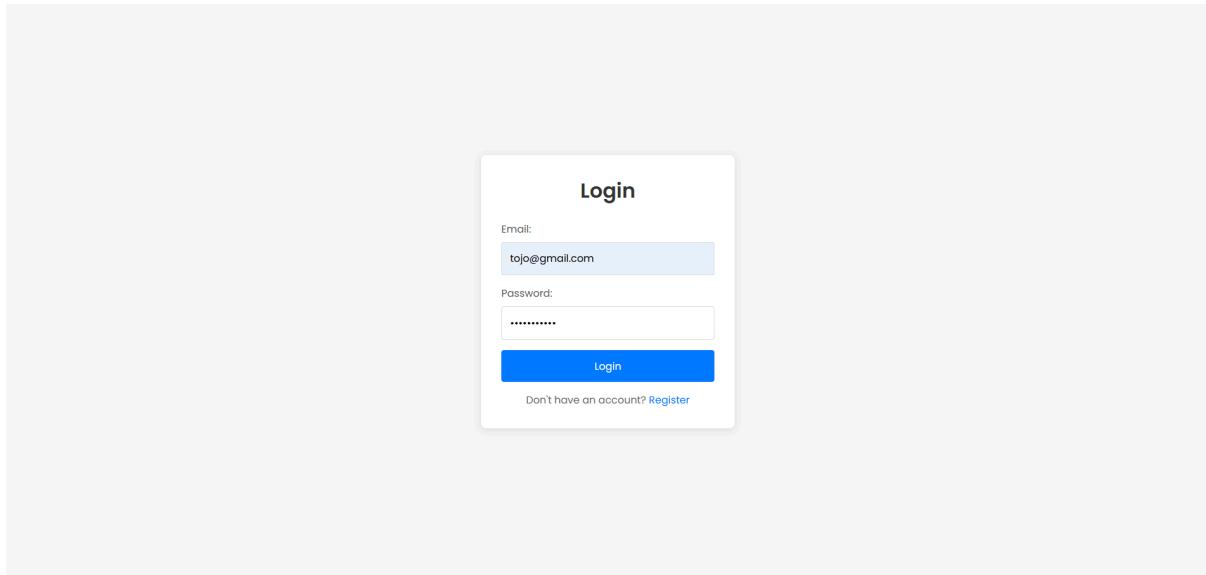
) : (
  <p>No items in your wishlist.</p>
)
</div>
)
}

```

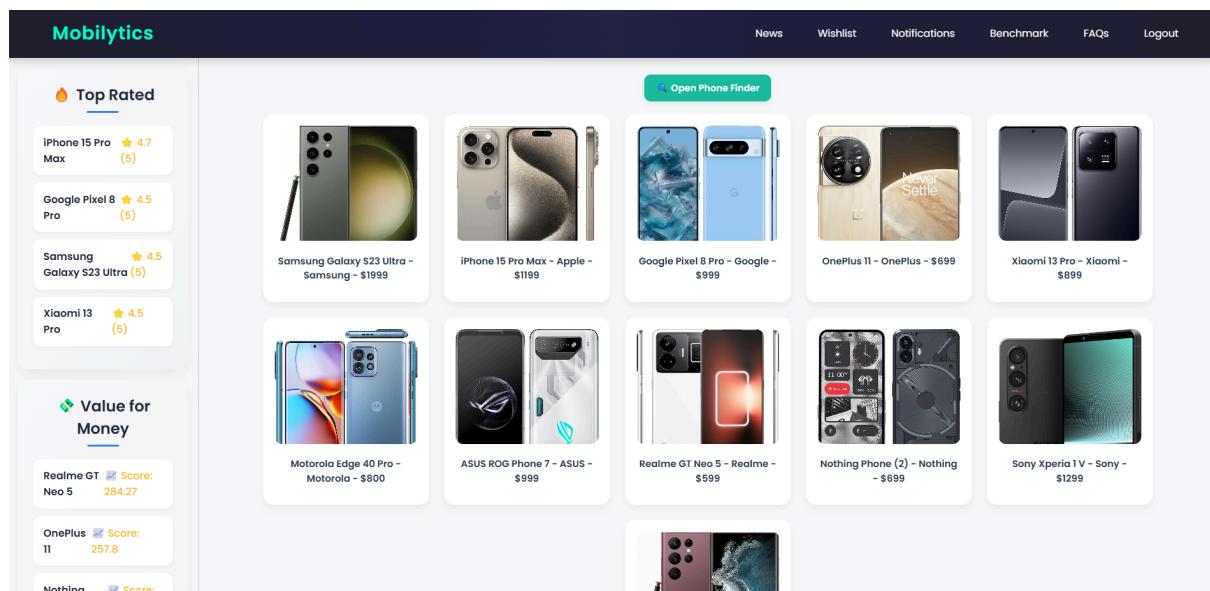
7. User Manual

For Normal User:

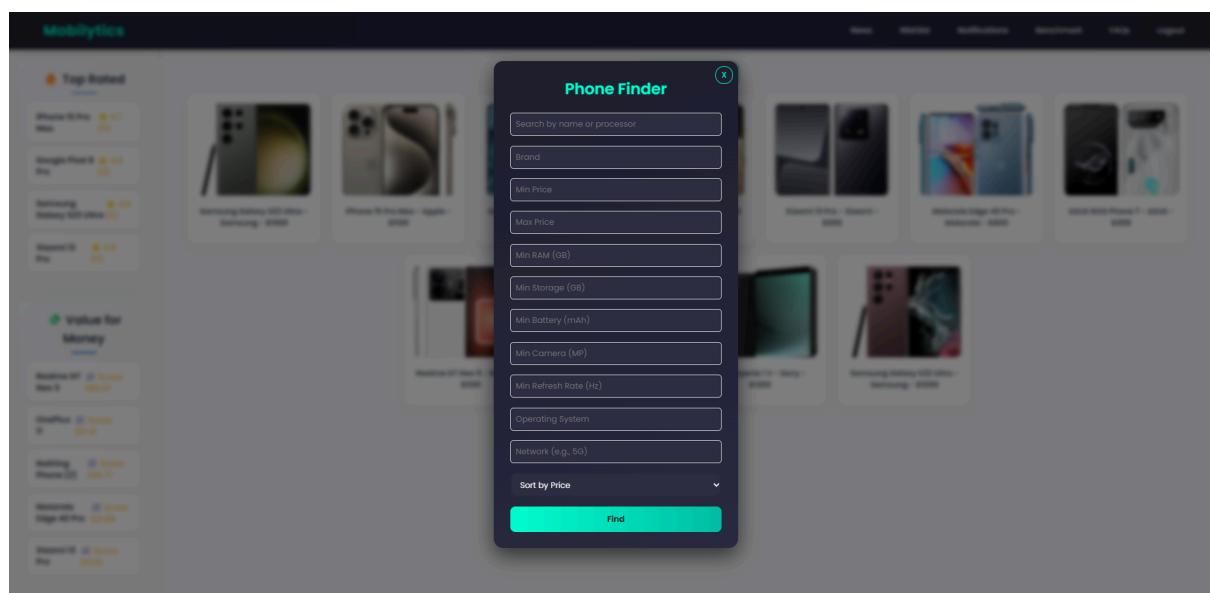
To process, the user must first log in. If the user does not have an account, they must register before logging in.



Once logged in, users will have access to a list of all available phones for browsing or further action. On the right side of the screen, there is a sidebar displaying the top 5 highest-rated phones and the top 5 best value-for-money phones.



Users can search, filter, and sort phones by entering their desired features using the *Phone Finder* tool.



Users can view detailed information about a phone by clicking on its entry in the list. On the phone details page, users will find a *Buy Now* button that redirects them to a third-party website for purchase. There is also an *Add to Wishlist* button, which adds the phone to the user's wishlist and enables price update notifications.

Mobilitytics

News Wishlist Notifications Benchmark FAQs Logout

Top Rated

iPhone 15 Pro Max	4.7
Google Pixel 8 Pro	4.5
Samsung Galaxy S23 Ultra	4.5
Xiaomi 13 Pro	4.5

Value for Money

Realme GT Neo 5	Score: 284.27
OnePlus 11	Score: 257.8
Nothing Phone (2)	Score: 240.77
Motorola Edge 40 Pro	Score: 221.88
Xiaomi 13 Pro	Score: 213.81

ASUS ROG Phone 7



Brand: ASUS
Price: \$999

[Buy Now](#) [Add to Wishlist](#)

Features

Camera	
Main	50 MP
UltraWide	13 MP
Front	32 MP
Video Recording	8K

Battery

Users can rate each phone and view ratings for individual features, such as performance, battery life, and camera quality.

Mobilitytics

News Wishlist Notifications Benchmark FAQs Logout

Average User Ratings
No ratings yet.

Submit Your Rating

CAMERA:

BATTERY:

DISPLAY:

PROCESSOR:

Comment:

[Submit Rating](#)

Network
5G, 4G, Wi-Fi 6E

SIM
Dual SIM

Dimensions
163.4 x 78.1 x 8.9 mm

Weight
234 grams

Mobilitytics

News Wishlist Notifications Benchmark FAQs Logout

Average User Ratings

Camera: 4.6 ★
Battery: 4.4 ★
Display: 4.6 ★
Processor: 4.8 ★
Total Ratings: 5

Rating Done!
Thank you for your feedback. You have already rated this product.

Network
5G, 4G, Wi-Fi 6E

SIM
Dual SIM

Dimensions
163.4 x 78.1 x 8.9 mm

Weight
234 grams

At the bottom of the phone details page, users can view a list of similar phones

The screenshot shows the 'Weight' section of a phone details page. It displays the weight as 213 grams. Below this is the 'Average User Ratings' section, which includes ratings for Camera (5.0), Battery (4.2), Display (4.6), Processor (4.4), and a total rating of 5. A message box says 'Rating Done!' and 'Thank you for your feedback. You have already rated this product.' Below this, there's a 'You might also like:' section featuring three phones: Xiaomi 13 Pro, ASUS ROG Phone 7, and Samsung Galaxy S22 Ultra, each with its price (\$899, \$999, and \$1099 respectively).

Users can access the latest news by clicking on the *News* section in the navigation bar.

The screenshot shows the 'Latest Phone News' section. On the left, there are two columns: 'Top Rated' and 'Value for Money'. The 'Top Rated' column lists the iPhone 15 Pro Max (4.7), Google Pixel 8 Pro (4.5), Samsung Galaxy S23 Ultra (4.5), and Xiaomi 13 Pro (4.5). The 'Value for Money' column lists the Realme GT Neo 5 (284.27), OnePlus 11 (257.8), Nothing Phone (2) (240.77), Motorola Edge 40 Pro (221.88), and Xiaomi 13 Pro (213.61). The main area features news articles with images and titles: 'Acer returns to smartphones with Super ZX and Super ZX Pro', 'Nvidia announces RTX 5060 and 5060 Ti graphics cards', and 'Google Pixel 9 First Look'.

Users can view their wishlist by clicking on the *Wishlist* section in the navigation bar.

The screenshot shows the 'Your Wishlist' section. It lists four phones: Samsung Galaxy S23 Ultra (\$1999), OnePlus 11 (\$699), and two instances of the Xiaomi 13 Pro (213.61). Each item has a 'Remove' button next to it.

Price update notifications for wishlist phones will appear in the *Notifications* section of the navigation bar.

Notifications

- iPhone 13 Pro price changed: \$5 → \$3 [Mark Read]
- iPhone 13 Pro price changed: \$50000 → \$5000 [Mark Read]
- iPhone 13 Pro price changed: \$50 → \$5000 [Mark Read]
- iPhone 13 Pro price changed: \$500 → \$50 [Mark Read]
- Galaxy A73 5G price changed: \$50 → \$500 [Mark Read]

Users can view frequently asked questions (FAQs) in the FAQ section and also post their own questions and answers.

Frequently Asked Questions

How to compare phones?

1. Go to Benchmark section from navbar
2. Select two phones
3. Click on 'Compare'

Post Your Question and Answer

Your Question
Enter your question here

Your Answer
Enter your answer here

Submit FAQ

Users can compare up to two phones and view their benchmark scores, strength analysis, and the winner by accessing the *Benchmark* section in the navigation bar.

Phone Benchmark

Select Phone A | Select Phone B | Compare

Benchmark Result

Motorola Edge 40 Pro	Google Pixel 8 Pro
69	73

Phone Benchmark

Motorola Edge 40 Pro | Google Pixel 8 Pro | Compare

Benchmark Result

Motorola Edge 40 Pro	Google Pixel 8 Pro
69	73

Feature Comparison

Feature	Motorola Edge 40 Pro	Google Pixel 8 Pro
Price	\$800	\$999
Performance Score	1240	1050
Processor	Snapdragon 8 Gen 2	Tensor G3
RAM	12 GB	12 GB
Storage	256 GB	256 GB
Battery	4600 mAh	5050 mAh
Display	pOLED	OLED

	Xiaomi 13 Pro	Motorola Edge 40 Pro
Score:	213.61	
Camera	Main: 50 MP, UltraWide: 50 MP, Front: 60 MP	Main: 50 MP, UltraWide: 48 MP, Front: 10.5 MP
Charging Speed	125	30
Cores	8	8
Clock Speed	3.2 GHz	2.9 GHz

WHY MOTOROLA EDGE 40 PRO WINS:

- ✓ Motorola Edge 40 Pro has better Performance.
- ✓ Google Pixel 8 Pro has better Battery.
- ✓ Motorola Edge 40 Pro has better Display.
- ✓ Motorola Edge 40 Pro has better ChargingSpeed.
- ✓ Motorola Edge 40 Pro has better ClockSpeed.

Finally the user can logout, pressing the logout button on the navigation bar.

For Admin User:

The admin user account will be predefined. The admin's navigation will be similar to that of regular users, but with additional options such as adding, editing, and deleting phones.

To delete a phone, the admin needs to click the *Delete* button.

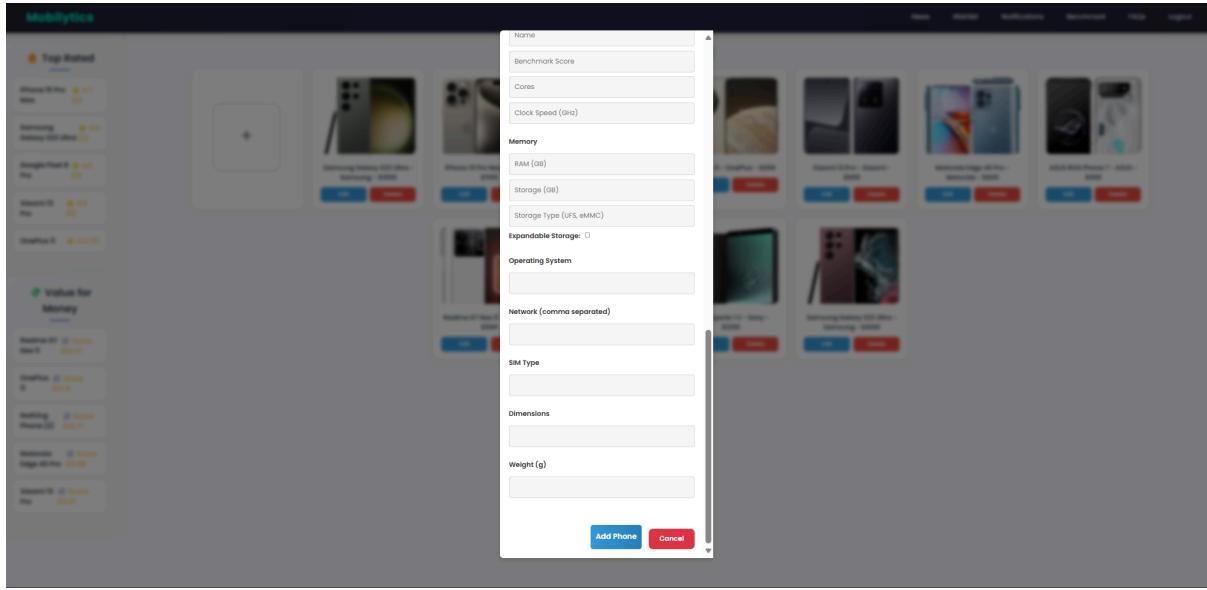
To add a phone, the admin user must click the plus icon, which will open a modal form. The admin then needs to enter the necessary information for the phone.

Top Rated

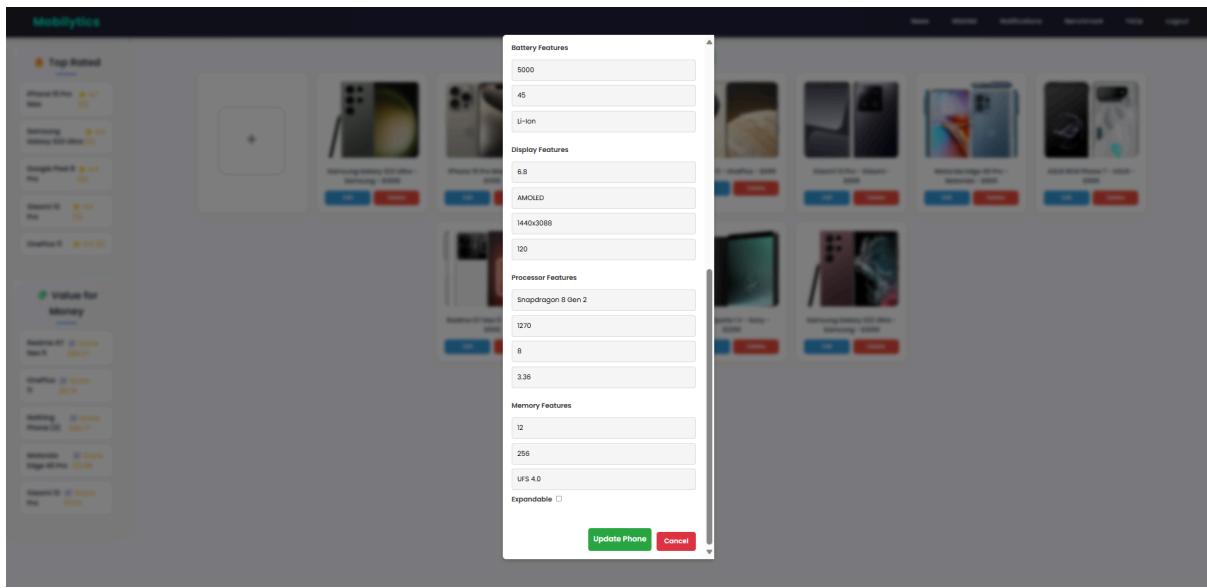
- iPhone 15 Pro Max ★ 4.7 (5) \$1999
- Samsung Galaxy S23 Ultra ★ 4.5 (5) \$1999
- Google Pixel 8 ★ 4.5 (5) \$999
- Xiaomi 13 Pro ★ 4.5 (5) \$899
- OnePlus 11 ★ 4.2 (5) \$899

Value for Money

- Realme GT Neo 5 ★ Score: 284.27
- OnePlus 11 ★ Score: 257.8
- Nothing Phone (2) ★ Score: 240.77
- Motorola Edge 40 Pro ★ Score: 221.88
- Xiaomi 13 Pro ★ Score: 213.61

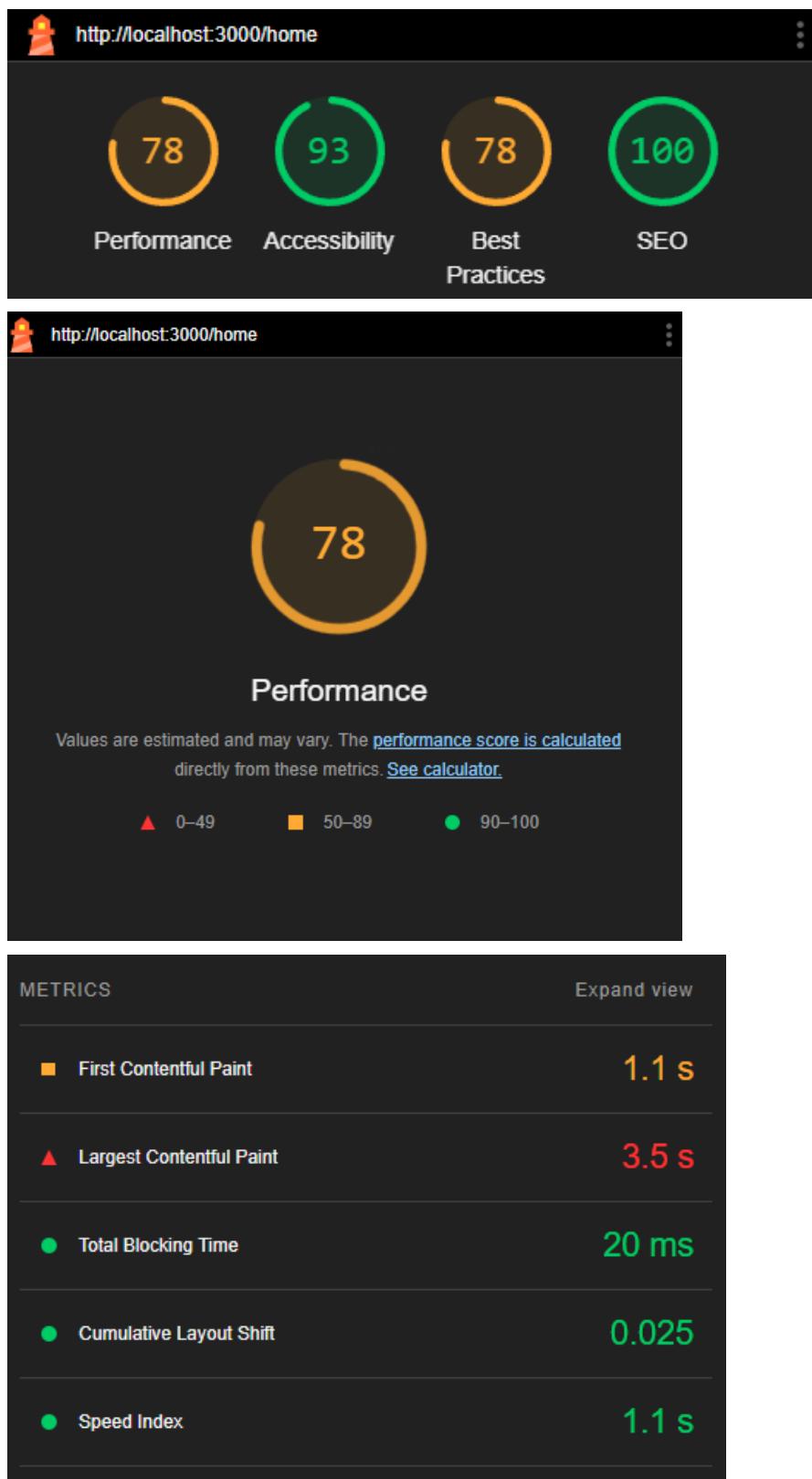


To update phone information, the admin needs to click the *Edit* button, which will open a modal form where the necessary details can be updated.



8. Performance and Network Analysis

Performance Score:



DIAGNOSTICS		
▲	Largest Contentful Paint element — 3,520 ms	▼
▲	Preload Largest Contentful Paint image — Potential savings of 470 ms	▼
▲	Minify JavaScript — Potential savings of 270 KiB	▼
▲	Eliminate render-blocking resources — Potential savings of 280 ms	▼
▲	Properly size images — Potential savings of 618 KiB	▼
▲	Reduce unused JavaScript — Potential savings of 279 KiB	▼
▲	Serve images in next-gen formats — Potential savings of 273 KiB	▼
▲	Preconnect to required origins — Potential savings of 110 ms	▼
▲	Page prevented back/forward cache restoration — 1 failure reason	▼
■	Minify CSS — Potential savings of 6 KiB	▼
■	Enable text compression — Potential savings of 19 KiB	▼

■	Serve static assets with an efficient cache policy — 1 resource found	▼
■	Remove duplicate modules in JavaScript bundles — Potential savings of 7 KB	▼
■	Avoid serving legacy JavaScript to modern browsers — Potential savings of 0 KiB	▼
○	Avoid large layout shifts — 13 layout shifts found	▼
○	Avoid non-composited animations — 2 animated elements found	▼
○	Avoid chaining critical requests — 3 chains found	▼
○	Minimize third-party usage — Third-party code blocked the main thread for 0 ms	▼
○	Avoid long main-thread tasks — 1 long task found	▼

Network Analysis:

The image shows two screenshots of a mobile application interface from 'Moblytics' and its corresponding Network tab in a browser developer tools window.

Top Rated

- iPhone 15 Pro Max** ★ 4.7 (5)
- Samsung Galaxy S23 Ultra** ★ 4.5 (5)
- Google Pixel 8 Pro** ★ 4.5 (5)
- Xiaomi 13 Pro** ★ 4.5 (5)
- OnePlus 11** ★ 4.2 (5)

Value for Money

iPhone 15 Pro Max ★ 4.7 (5)

Samsung Galaxy S23 Ultra ★ 4.5 (5)

Google Pixel 8 Pro ★ 4.5 (5)

Xiaomi 13 Pro ★ 4.5 (5)

OnePlus 11 ★ 4.2 (5)

Buy Now

Add to Wishlist

iPhone 15 Pro Max

Brand: Apple

Price: \$1199

Buy Now

Add to Wishlist

Network Tab (Top Screenshot)

Request URL: <http://localhost:1000/api/rating/68080a2142d0811957f1b4d4>

Request Method: GET

Status Code: 200 OK

Remote Address: [::1]:1080

Referrer Policy: strict-origin-when-cross-origin

Access-Control-Allow-Origin: *

Content-Length: 113

Content-Type: application/json; charset=utf-8

Date: Fri, 02 May 2025 17:07:54 GMT

Etag: W/72wQSENTRVAgQgQF9YEDTROE

X-Powered-By: Express

Request Headers

Accept: application/json, text/plain, */*

Accept-Encoding: gzip, deflate, br, zstd

Accept-Language: en-US,en;q=0.9

Connection: keep-alive

Host: localhost:1080

If-None-Match: W/72wQSENTRVAgQgQF9YEDTROE

Origin: <http://localhost:3000>

Sec-Ch-Ua: "Google Chrome";v="135", "Not-A-Brand";v="8", "Chromium";v="135"

Sec-Ch-Ua-Mobile: 70

Sec-Ch-Ua-Platform: "Windows"

Console: All assistance ⓘ What's new Network conditions

Default levels ⓘ 2 issues ⓘ

Network Tab (Bottom Screenshot)

Request URL: <http://localhost:1000/api/rating/68080a2142d0811957f1b4d4>

Request Method: GET

Status Code: 200 OK

Remote Address: [::1]:1080

Referrer Policy: strict-origin-when-cross-origin

Access-Control-Allow-Origin: *

Content-Length: 113

Content-Type: application/json; charset=utf-8

Date: Fri, 02 May 2025 17:07:54 GMT

Etag: W/72wQSENTRVAgQgQF9YEDTROE

X-Powered-By: Express

Request Headers

Accept: application/json, text/plain, */*

Accept-Encoding: gzip, deflate, br, zstd

Accept-Language: en-US,en;q=0.9

Connection: keep-alive

Host: localhost:1080

If-None-Match: W/72wQSENTRVAgQgQF9YEDTROE

Origin: <http://localhost:3000>

Sec-Ch-Ua: "Google Chrome";v="135", "Not-A-Brand";v="8", "Chromium";v="135"

Sec-Ch-Ua-Mobile: 70

Sec-Ch-Ua-Platform: "Windows"

Console: All assistance ⓘ What's new Network conditions

Default levels ⓘ 2 issues ⓘ

Mobile ViewPort:

Mobilytics News Wishlist Notifications Benchmark FAQs Logout

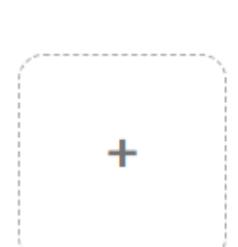
Top Rated

- iPhone 15 Pro Max ★ 4.7 (5)
- Google Pixel 8 Pro ★ 4.5 (5)
- Samsung Galaxy S23 Ultra ★ 4.5 (5)
- Xiaomi 13 Pro ★ 4.5 (5)
- OnePlus 11 ★ 4.2 (5)

Value for Money

- Realme GT Neo 5 Score: 284.27
- OnePlus 11 Score: 257.8
- Nothing Phone (2) Score: 240.77
- Motorola Edge 40 Pro Score: 221.88
- Xiaomi 13 Pro Score: 213.61

Open Phone Finder





Samsung Galaxy S23 Ultra - Samsung - \$1999

Edit Delete



iPhone 15 Pro Max - Apple - \$1199

Edit Delete



Google Pixel 8 Pro - Google - \$999

Edit Delete



OnePlus 11 - OnePlus - \$699

Edit Delete



Xiaomi 13 Pro - Xiaomi - \$899

Edit Delete

9. Github Repo [Public] Link:
[EbrerRamen/Mobile-Benchmarking-System](#)

10. Link of Deployed Project: Not deployed

11. Individual Contribution

Group member - 01	
Name: Md. Abrar Rahman Shafin	Student ID: 21201080
Functional Requirements which are developed by this member:	
1. Phone Database Management (Admin Panel)	
2. Phone Specification Display	
3. Search and Filter	
4. Comparison	
5. Benchmark	
6. Top 5 Value for Money Phones	

Group member - 02	
Name: Yasin Rahman	Student ID: 23341115
Functional Requirements which are developed by this member:	
1. User Authentication	
2. News and Update Section	
3. FAQ	
4. User Scored Feature Ratings	
5. Top 5 highest Rated Phones	

Group member - 03	
Name: Torsha Bashar	Student ID: 21301083
Functional Requirements which are developed by this member:	
1. Phone Purchase Option	
2. Save Phones to Wishlist	
3. Price Update Notification	
4. User reviews	

12. References

- a. [GSMArena.com - mobile phone reviews, news, specifications and more...](#)
- b. [Home - UserBenchmark](#)