

# Title: Pocket Forecaster

## Roadmap:

An AI Powered Smartphone Recommendation System

**Status :** Project fully completed, including Java backend, MVC architecture, Strategy Pattern logic, NLP integration, and final testing.

**Data Format :** CSV

**AI Model :** VADER + BERT (Hybrid model For Sentiment Analysis of the user **reviews**)

### Abstract:

Pocket Forecaster is a smart smartphone recommendation system built in Java that suggests the best mobile phones based on a user's budget, usage type, and operating system preference. It is designed with clean OOP structure, MVC architecture, and strategy driven design patterns to keep the system organized, scalable, and easy to update. The platform also uses a hybrid sentiment analysis model combining VADER and BERT to analyze real customer reviews and give users clear insights into how people feel about each device. Along with phone suggestions, it recommends useful add-ons like accessories, apps, or games based on the user's usage style. All results come from well-processed CSV datasets and are shown through a simple, responsive web interface, making Pocket Forecaster a practical and user-friendly tool that helps people choose the right smartphone with confidence.

### Problem Solved :

- Too many smartphone choices
- Scattered and biased reviews
- Usage–feature mismatch
- No unified comparison system

### Solution :

Smart Filtering

AI sentiment for real reviews

Direct Store Link

Get users actual use case of a mobile and recommends

### Key Features:

1) Smart Filtering = Budget + Usage(Gaming/Photography/Casual/etc)+OS

2) AI Sentiment Analysis = Get sentiment from the reviews (For instance :

89% positive and generate a one line review for the displayed mobile)

3) Personalized Add-on - Accessories that is suggested along with the recommendations.

4) Visually Appealing UI = Responsive HTML / CSS and JavaScript.

5) Strategy Driven design = Strategy driven design for making feature logic more flexible.

6) Model View Controller = MVC architecture pattern to design the outline of the project.

7) Junit Test = Unit testing is done for classes and methods in it for correctness and proper working.

## **Tech Stack:**

### **Language & Backend Development**

- Java (OOP, Streams API)
- MVC Architecture
- Strategy Design Pattern
- Spark Java API

### **Frontend**

- HTML, CSS, JavaScript

### **Data & Storage**

- Real-world smartphone datasets stored in CSV
- Apache Commons CSV for parsing and preprocessing

### **AI / NLP Integration**

- Python (NLP – Hybrid Sentiment Analysis using VADER + BERT)

### **System Design**

- Modular OOP structure
- Strategy-driven filtering engine
- MVC-based separation of concerns

### **Tech Stack Usage :**

Java (70%) = MVC , Strategy design , Filtering , Batch navigation logic , Add-on suggestion .

HTML, CSS , JavaScript (20%) = For the UI

CSV data (7%) = Store the processed data

CSV Parser (2%) = converting CSV file into java Objects

VADER AI (1%) = Runs to generate Sentiment Score for reviews.

## Data Flow:

### 1. CSV Load (Start)

All pre-processed CSV files (specifications, reviews, sentiment scores, add-ons) are loaded into memory for fast filtering.

### 2. User Input

User enters:

- Budget range
- Usage type
- OS preference

### 3. Processing

- Java filters phones based on budget, usage, and OS
- Strategy Pattern applies usage-specific filtering logic
- Results sorted by: Sentiment score → Price → Usage relevance → OS match

### 4. Output

- Top 3 **best-fit phones** displayed with sentiment score, short review summary, and “Buy Now” link
- Add-on suggestions shown (accessories, apps, bonuses based on usage)

### 5. Additional Options

If the user wants more results, a “next batch” of phones (including slightly higher-budget premium options) is displayed.

## AI Sentiment Pipeline:

raw\_phones.csv

↓

[Python + VADER and BERT] → Run ONCE

↓

phones\_with\_sentiment.csv

↓

Java displays: "89% Positive"

# Project Timeline

Week	Milestone
<b>Oct 20–30</b>	Core Java logic with test cases.
<b>Nov 1–10</b>	Web UI, responsive design, CSV, filtering, AI sentiment
<b>Nov 11–20</b>	Add-ons, batch navigation
<b>Nov 21–30</b>	Testing, Javadoc, polish
<b>Dec 1–3</b>	Final demo

## Example Web output:

Budget: ₹80,000

Usage: Gaming

OS: Android

1) ASUS ROG Phone 8 – ₹74,999

"Built for heavy gaming"

Sentiment: 89% Positive (487 reviews)

[Buy → <https://www.91mobiles.com/asus-rog-phone-8-price-in-india> ]

2) OnePlus 12 – ₹59,999

Sentiment: 84% Positive

3) Samsung Galaxy S23 – ₹69,999

Sentiment: 90% Positive

## Bonus Suggestions (Gaming Enthusiast Edition)

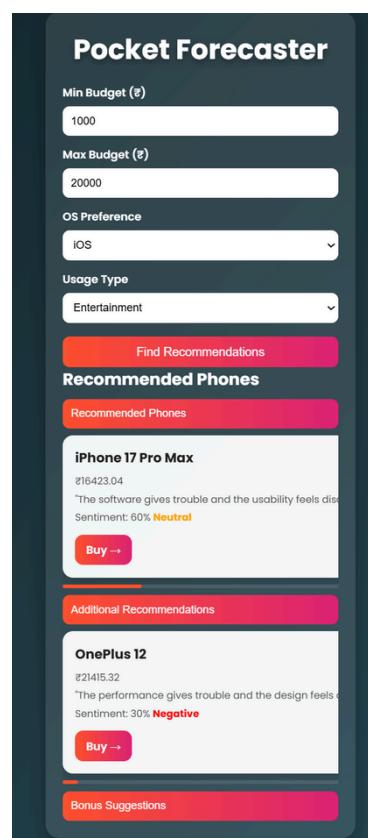
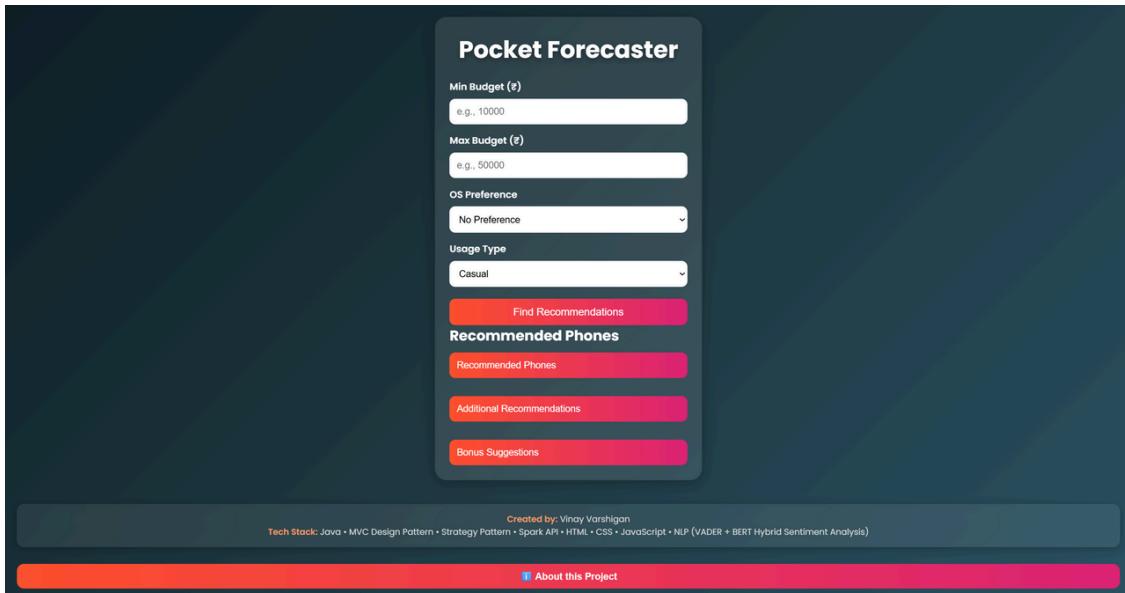
### 🎮 Top Gaming Accessories:

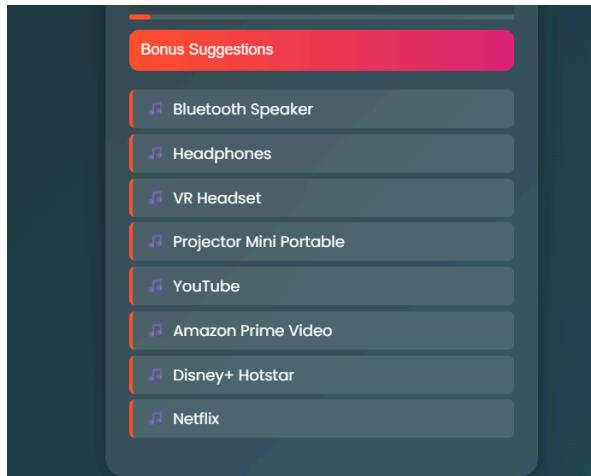
- Razer Kishi Mobile Game Controller
- Black Shark Cooling Fan
- JBL Quantum TWS Earbuds

## 🔥 Trending Mobile Games:

- Call of Duty: Mobile
- Asphalt 9: Legends
- PUBG: New State

## Output Screenshots:





## Conclusion and Future work:

Pocket Forecaster shows how software engineering and AI can work together to create a smart and user-friendly smartphone recommendation system. With modular OOP, MVC, and the Strategy Pattern, the system stays flexible, scalable, and easy to maintain. Real customer opinions are captured through VADER sentiment analysis, making the suggestions more practical than basic specs-only comparisons. Features like add-on recommendations and next-batch navigation improve usability and personalization. Overall, Pocket Forecaster is an effective, data-driven tool that helps users choose the right smartphone with confidence and ease.

## Future Work

- **Real-time Pricing:** Integrate APIs to auto-update phone prices, ratings, and availability.
- **User Accounts:** Add login, saved preferences, and personalized recommendation history.
- **ML Ranking Model:** Build a model to score phones based on performance, battery, camera, and sentiment.
- **Advanced NLP:** Upgrade from VADER to transformer models(GPT-Based Sentiment Models/RoBERTa) for deeper review analysis.
- **Smart Add-ons:** Auto-generate accessory/app suggestions using ML instead of static CSV mapping.
- **UI/UX Improvements:** Develop a mobile-friendly interface with smoother design and dark mode.
- **Country Specific Pricing:** Enable users to select their country/currency and automatically convert phone prices using exchange-rate multipliers or real-time forex APIs for region-accurate recommendations.
- **Hybrid Recommender:** Combine content-based and collaborative filtering for smarter results.
- **Cloud Deployment:** Deploy on AWS/Azure with database support, CI/CD, and monitoring.

