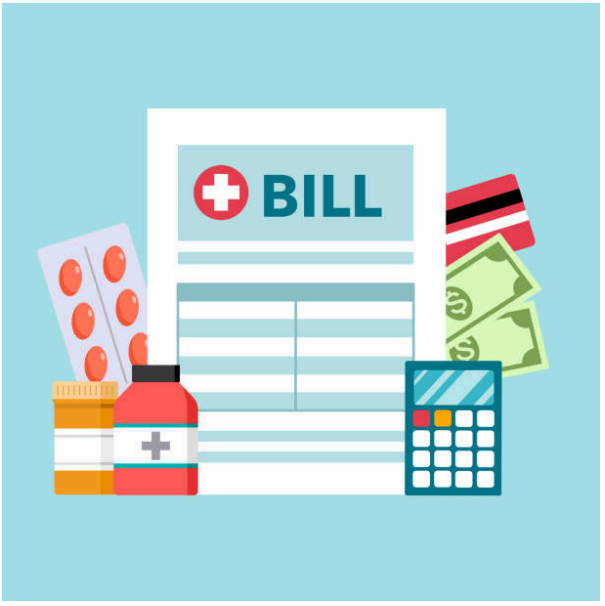
GenMedi App: AI-Powered Drug Alternatives

By- Kunal Bhure

Product Description

The **AI-integrated app** helps users identify **alternative generic medicines** for prescribed branded drugs. This is particularly useful for patients seeking cost-effective options and for those looking for substitutes due to availability issues.

The app leverages advanced machine learning and AI to provide users with a list of equivalent generic drugs based on a comprehensive pharmaceutical database. GenMedi App empowers patients to make informed decisions, **reduces healthcare costs**, and supports healthcare providers in their prescription practices, aiming to create a more inclusive and equitable healthcare system.



Business Model and Monetization Strategy

According to the analysis, a major source of income will be through mobile applications. Various pricing models can be applied. Also, we can earn through various in-app purchases. According to our speculation, we can have four major sources of income for our product. Details about them are as follows:

Freemium Model:

- **Free Tier:** Users can search for generic alternatives for up to 5 branded drugs per month.
- **Premium Tier:** Users pay a subscription fee for unlimited searches, advanced search options (e.g., filtering by side effects, manufacturers, or specific compounds), and access to a database of reviews and ratings for different generic alternatives.

Pharmaceutical Partnerships:

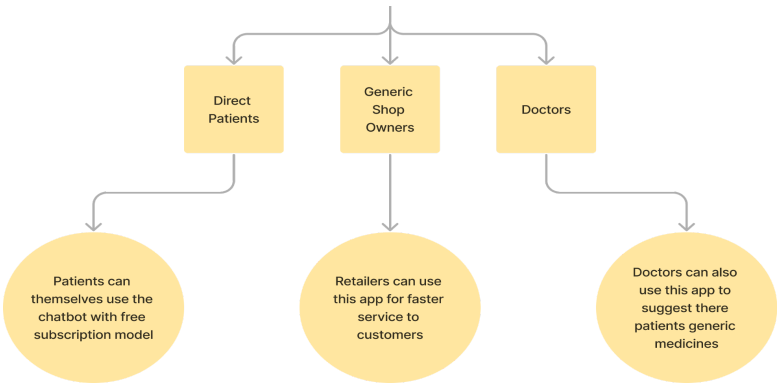
- Partner with pharmaceutical companies to provide insights or advertisements within the app. These companies could pay for advertising space or to promote specific generic alternatives.

In-App Purchases:

- Offer detailed reports, drug interaction checks, or personalized medicine recommendations as one-time purchases within the app.

Data Analytics for Healthcare Providers:

- Aggregate anonymized data on drug substitution trends and sell it to healthcare providers or research institutions for a fee.



Financial Equation

Given the business model, the financial equation can be developed based on the revenue from subscriptions, partnerships, in-app purchases, and data sales.

Let:

- Rs = Revenue from subscriptions
- Rp = Revenue from pharmaceutical partnerships
- Ri = Revenue from in-app purchases
- Rd = Revenue from data analytics
- Cf = Fixed costs (development, maintenance, etc.)
- Cv = Variable costs (marketing, customer support, etc.)

The financial equation for profit PPP can be expressed as:

$P=(Rs+Rp+Ri+Rd)-(Cf+Cv)$

Where:

- $Rs=Ns\times Ps$ with NsN_sNs being the number of subscribers and PsP_sPs the price per subscription.
- Rp is a fixed amount based on contracts with pharmaceutical companies.
- $Ri=Ni\times Pi$ where NiN_iNi is the number of in-app purchases and PiP_iPi the price per purchase.
- Rd is a variable amount based on data sales agreements.
- Cf includes development costs, server maintenance, etc.
- Cv includes marketing, customer acquisition costs, etc.

Let's take a guesstimate for our product,

Revenue Streams Breakdown:

1. **Revenue from Subscriptions (R_s):**

- **Estimated Number of Subscribers (N_s):** Let's assume that 2% of the total smartphone users in India are potential subscribers.
- **Smartphone Users in India:** Approximately 600 million.
- **Potential Subscribers:** $N_s = 600 \times 106 \times 0.02 = 12 \times 106$ (12 million)
- **Subscription Price (P_s):** ₹150 per month.
- **Annual Revenue from Subscriptions:** $Rs = Ns \times Ps \times 12 \approx 12 \times 106 \times 150 \times 12 = 21.6 \times 109$ (₹21.6 billion)

2. **Revenue from Pharmaceutical Partnerships (R_p):**

- **Estimated Number of Partners:** Assume partnerships with 50 pharmaceutical companies.
- **Annual Revenue per Partner:** ₹10 million.
- **Total Annual Revenue from Partnerships:** $Rp = 50 \times 10 \times 106 = 0.5 \times 109$ (₹500 million)

3. **Revenue from In-App Purchases (R_i):**

- **Estimated Number of In-App Purchases (N_i):** Assume 5% of free-tier users make in-app purchases.
- **Potential Free-Tier Users:** Assume 10% of total smartphone users = 60 million.
- **In-App Purchase Price (P_i):** ₹50 per purchase.
- **Total Annual Revenue from In-App Purchases:** $Ri = Ni \times Pi = 0.05 \times 60 \times 106 \times 50 \approx 150 \times 106$ (₹150 million)

4. **Revenue from Data Analytics (R_d):**

- **Estimated Revenue from Data Sales:** Assume ₹1 billion annually from selling anonymized data to healthcare providers and research institutions.
- **Total Annual Revenue from Data Analytics:** $Rd = 1 \times 109$ (₹1 billion)

Cost Structure:

1. **Fixed Costs (C_f):**
    - **App Development:** ₹100 million (one-time).
    - **Server and Maintenance Costs:** ₹50 million annually.

- **Total Fixed Costs:** $C_f = 100 \times 10^6 + 50 \times 10^6 = 150 \times 10^6$ (₹150 million)

2. **Variable Costs (C_v):**
   - **Marketing and Advertising:** ₹200 million annually.
   - **Customer Support and Operations:** ₹100 million annually.
   - **Total Variable Costs:** $C_v = 200 \times 10^6 + 100 \times 10^6 = 300 \times 10^6$ (₹300 million)

Total Revenue Calculation:

$$\text{Total Revenue } (R) = R_s + R_p + R_i + R_d$$

$$R \approx 21.6 \times 10^9 + 0.5 \times 10^9 + 0.15 \times 10^9 + 1 \times 10^9 \approx 23.25 \times 10^9 \text{ (₹23.25 billion)}$$

Total Cost Calculation:

$$\text{Total Costs } (C) = C_f + C_v = 150 \times 10^6 + 300 \times 10^6 = 450 \times 10^6 \text{ (₹450 million)}$$

Net Profit Calculation:

$$\text{Net Profit } (P) = \text{Total Revenue } (R) - \text{Total Costs}(C)$$

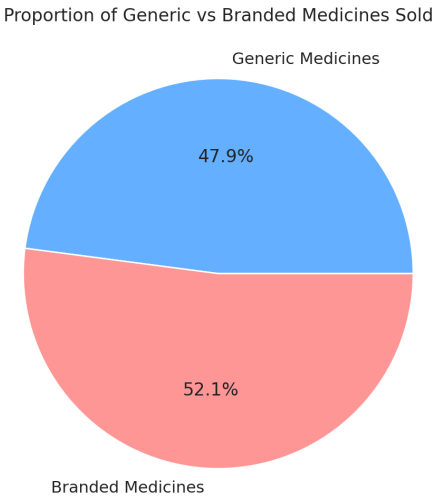
$$P = 23.25 \times 10^9 - 450 \times 10^6 = 22.8 \times 10^9 \text{ (₹22.8 billion)}$$

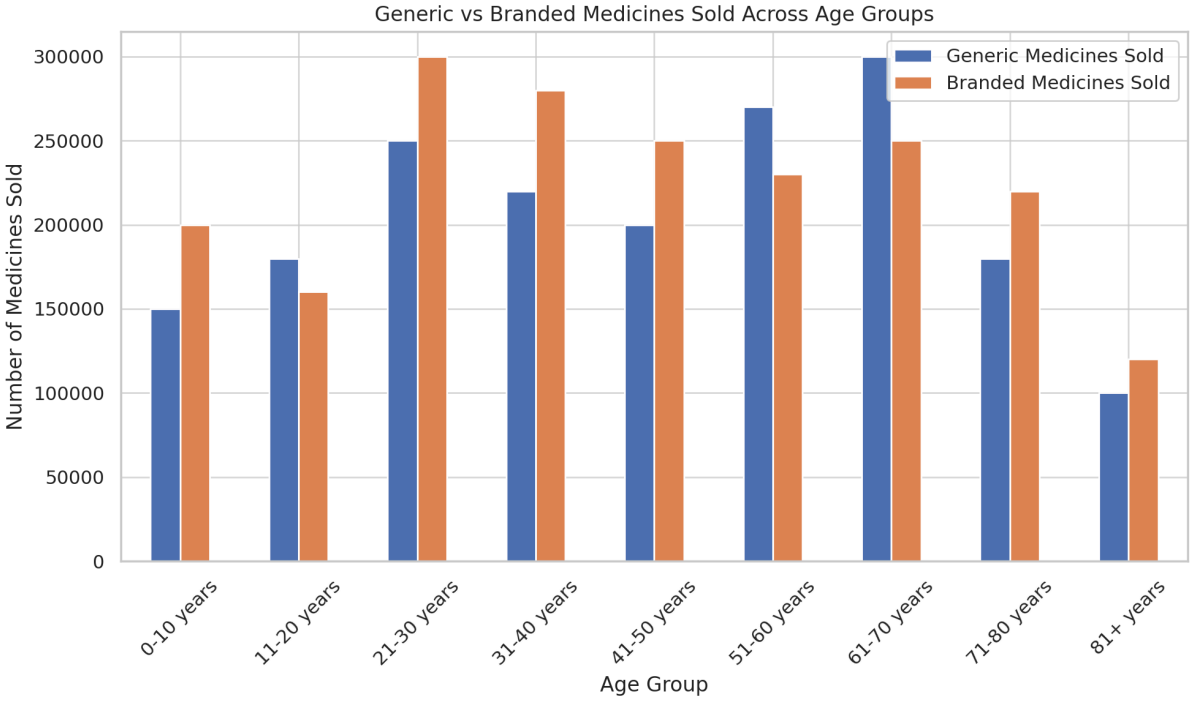
Summary:

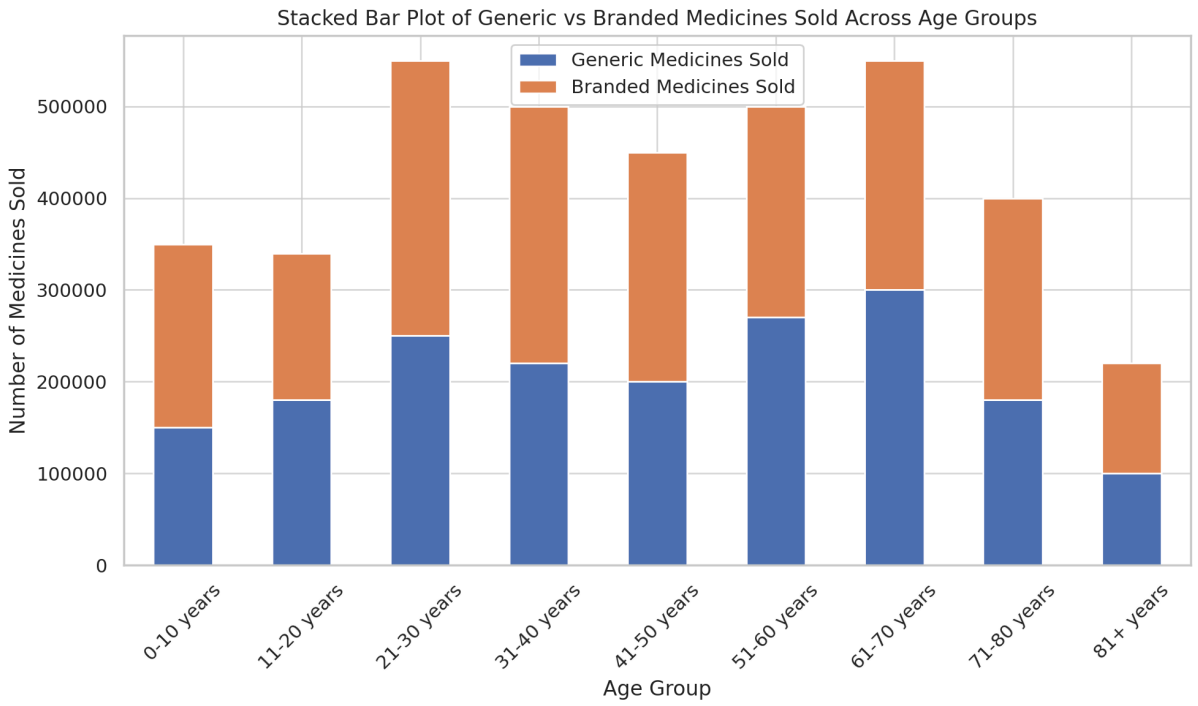
- **Total Estimated Annual Revenue:** ₹23.25 billion.
- **Total Estimated Costs:** ₹450 million.
- **Estimated Net Profit:** ₹22.8 billion.

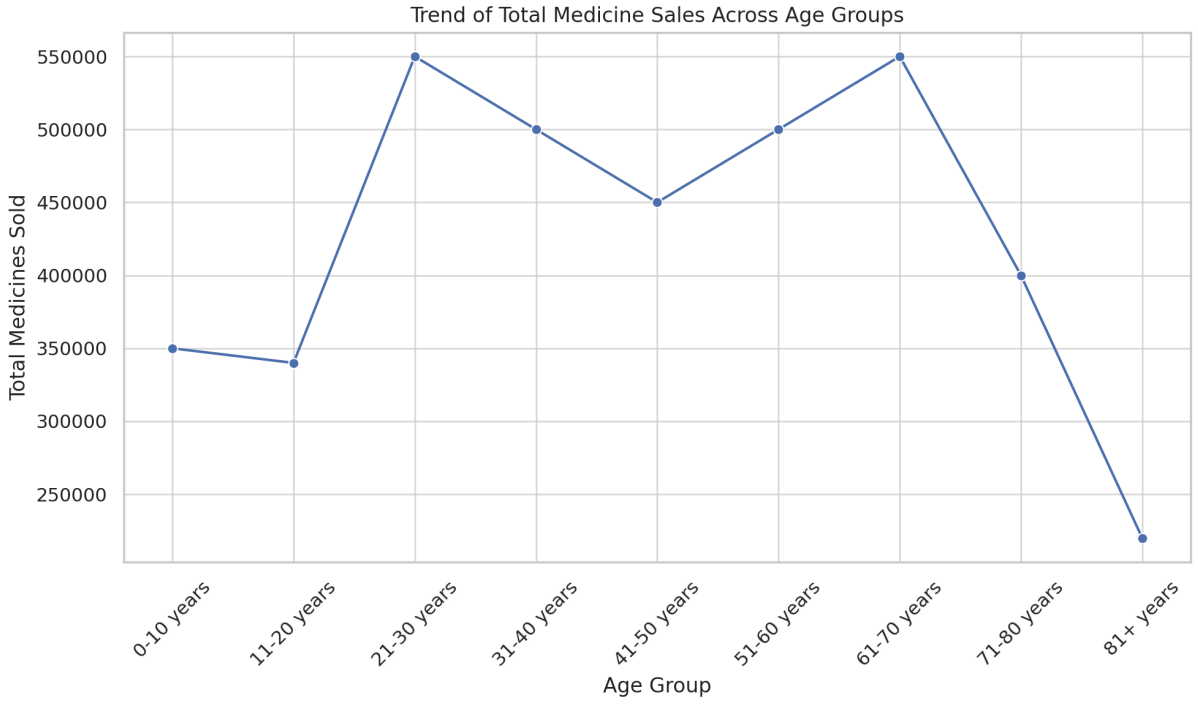
Data Visualisation

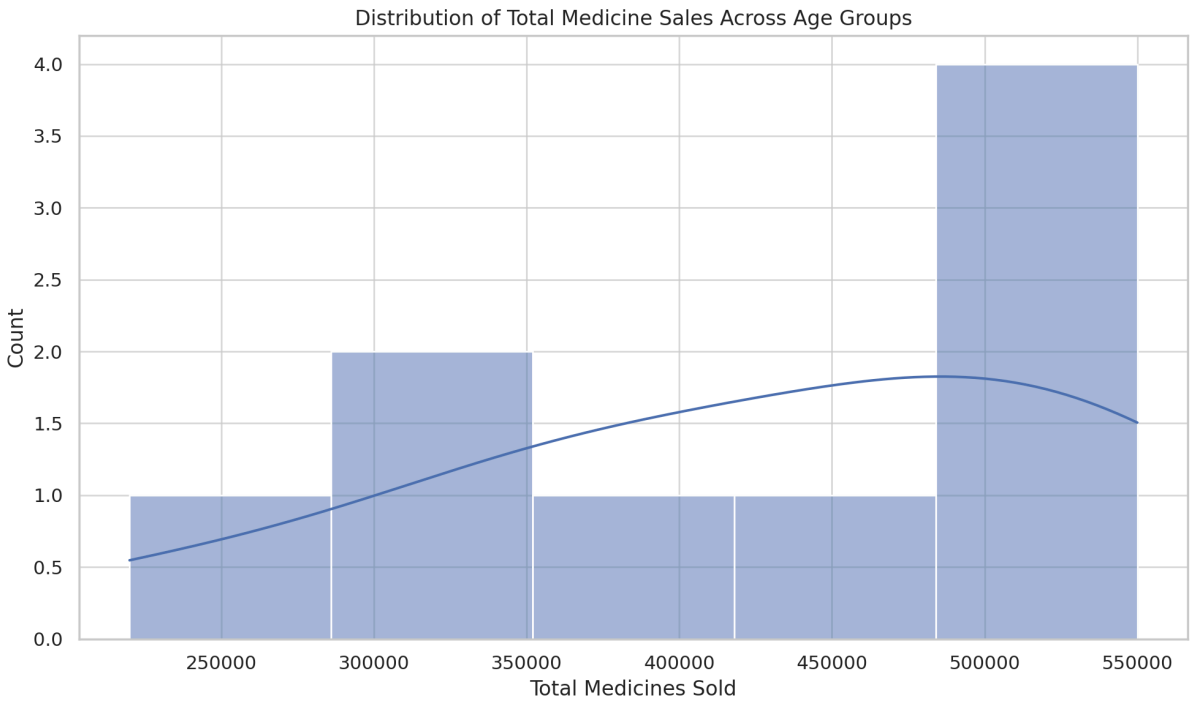
Here are some of the plots relevant to the product.

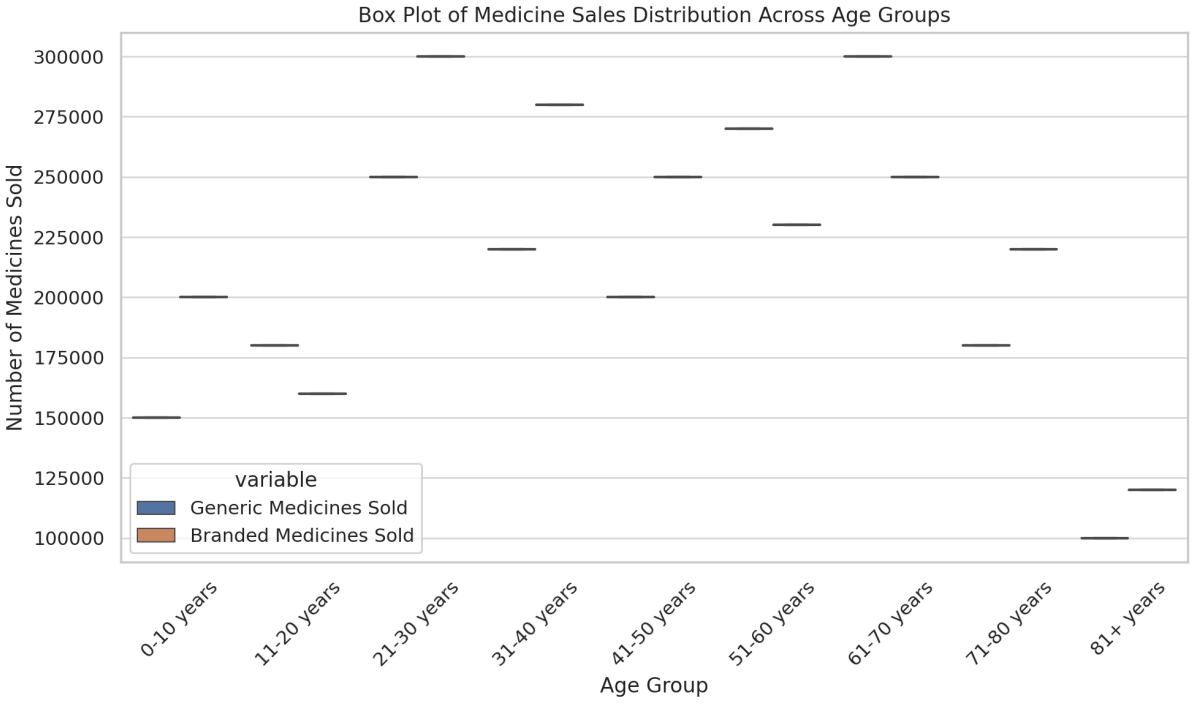





Stacked Bar Plot of Generic vs Branded Medicines Sold Across Age Groups
Generic Medicines Sold
Branded Medicines Sold
Number of Medicines Sold
0
100000
200000
300000
400000
500000
0-10 years
11-20 years
21-30 years
31-40 years
41-50 years
51-60 years
61-70 years
71-80 years
81+ years
Age Group


Trend of Total Medicine Sales Across Age Groups
Total Medicines Sold
250000
300000
350000
400000
450000
500000
550000
0-10 years
11-20 years
21-30 years
31-40 years
41-50 years
51-60 years
61-70 years
71-80 years
81+ years
Age Group


Distribution of Total Medicine Sales Across Age Groups
Count
Total Medicines Sold
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
250000
300000
350000
400000
450000
500000
550000


Box Plot of Medicine Sales Distribution Across Age Groups
Number of Medicines Sold
Age Group
variable
Generic Medicines Sold
Branded Medicines Sold
100000
125000
150000
175000
200000
225000
250000
275000
300000
0-10 years
11-20 years
21-30 years
31-40 years
41-50 years
51-60 years
61-70 years
71-80 years
81+ years

Market Segmentation

Target Market:

- **Primary Market:** Individuals who regularly purchase medications, particularly those looking for cost-effective alternatives or who are uninsured.
- **Secondary Market:** Healthcare providers and pharmacists who want to offer patients alternative options.
- **Tertiary Market:** Pharmaceutical companies are interested in trends related to drug substitution.

Segmentation Criteria:

- **Demographic:** Age (18-65), Income level (middle to low-income households), Uninsured or underinsured individuals.
- **Behavioral:** Frequent users of prescription medication, price-sensitive consumers, and individuals who prefer online healthcare solutions.
- **Geographic:** Focus on urban areas with higher access to the internet and smartphone usage.

From our analysis, we can infer that the age group 41-70 years has the largest amount of medicines sold. So they are our target customers.
No specific geographic pattern was observed thus concluding even distribution in different states of India.

Prototype Development:

A simple mobile app prototype can be developed using Flutter or React Native. The app will feature:

- A search bar where users can input the name of a branded drug.
- An AI model (integrated via an API) that processes the input and returns a list of possible generic alternatives.
- Basic UI elements to display the results, with options to filter based on user preferences.

Basic Python code structure for the core functionalities of our AI-integrated medicine substitute app. This code focuses on:

1. **Search functionality** for branded drugs.
2. **AI model integration** to suggest generic alternatives.
3. **Filtering functionality** based on user preferences

```python
from flask import Flask, request, jsonify

app = Flask(__name__)

# Sample drug database (for demonstration)
drug_database = {
    'Aspirin': ['GenericA', 'GenericB', 'GenericC'],
    'Tylenol': ['GenericD', 'GenericE'],
    # Add more drugs and their generic alternatives here
}

# Sample pricing and other metadata (for demonstration)
drug_info = {
    'GenericA': {'price': 10, 'manufacturer': 'Pharma1'},
    'GenericB': {'price': 12, 'manufacturer': 'Pharma2'},
    'GenericC': {'price': 8, 'manufacturer': 'Pharma3'},
    'GenericD': {'price': 15, 'manufacturer': 'Pharma1'},
    'GenericE': {'price': 20, 'manufacturer': 'Pharma4'},
    # Add more details as needed
}
```

```python
def search_drug():
    branded_drug = request.args.get('drug_name')
    if branded_drug in drug_database:
        alternatives = drug_database[branded_drug]
        return jsonify({'alternatives': alternatives})
    else:
        return jsonify({'message': 'Drug not found'}), 404
```

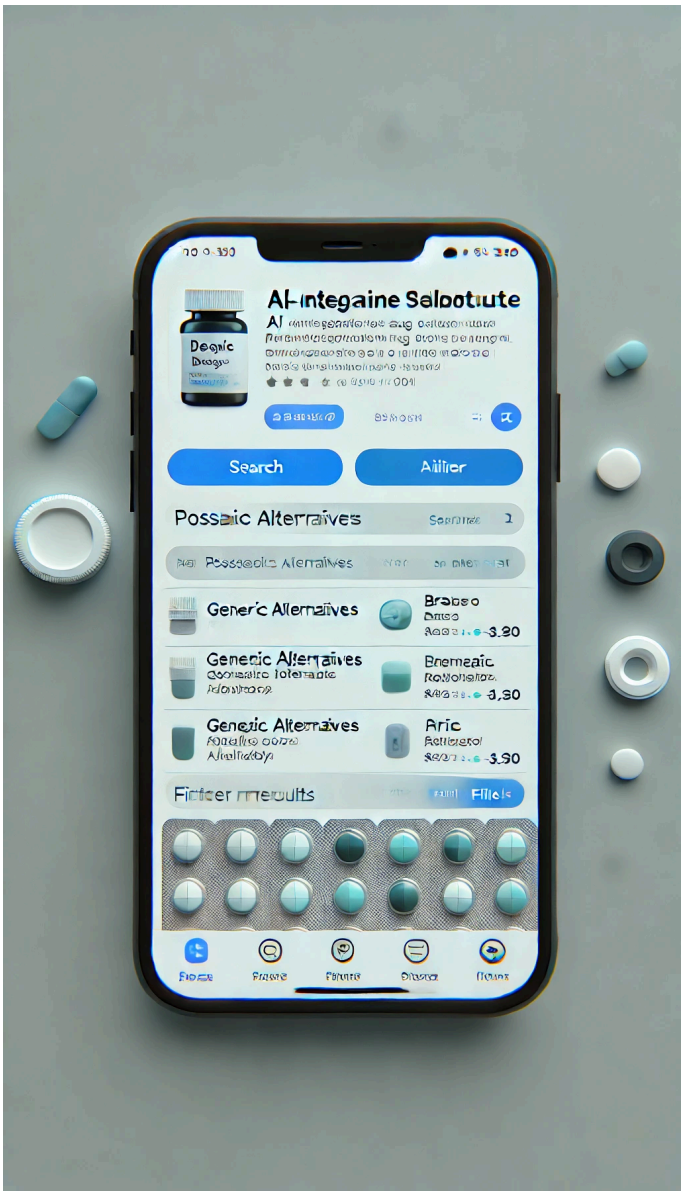
```python
def ai_model_select(alternatives):
    # Simulated AI model that ranks alternatives
    scored_alternatives = sorted(alternatives, key=lambda x: drug_info[x]['price'])
    return scored_alternatives

@app.route('/get_alternatives', methods=['GET'])
def get_alternatives():
    branded_drug = request.args.get('drug_name')
    if branded_drug in drug_database:
        alternatives = drug_database[branded_drug]
        selected_alternatives = ai_model_select(alternatives)
        return jsonify({'selected_alternatives': selected_alternatives})
    else:
        return jsonify({'message': 'Drug not found'}), 404
```

```python
@app.route('/filter', methods=['GET'])
def filter_alternatives():
    branded_drug = request.args.get('drug_name')
    filter_by = request.args.get('filter_by')  # e.g., 'price' or 'manufacturer'

    if branded_drug in drug_database:
        alternatives = drug_database[branded_drug]
        if filter_by == 'price':
            filtered = sorted(alternatives, key=lambda x: drug_info[x]['price'])
        elif filter_by == 'manufacturer':
            manufacturer = request.args.get('manufacturer')
            filtered = [alt for alt in alternatives if drug_info[alt]['manufacturer'] == manufacturer]
        else:
            filtered = alternatives
        return jsonify({'filtered_alternatives': filtered})
    else:
        return jsonify({'message': 'Drug not found'}), 404
```

The basic UI of our app is depicted as shown in the below image



Conclusion

This report outlines the key components necessary to bring the AI-integrated app to market, including its business model, financial structure, market segmentation, and a small-scale implementation plan.