

A Finance Managing Application “FINOVA”

1. FARZAAN KHAN B.Tech
IIMT COLLEGE OF
ENGINEERING GREATER
NOIDA ,201310

farzaank5@gmail.com

2. UTKARSH SAXENA B.Tech
IIMT COLLEGE OF
ENGINEERING GREATER
NOIDA ,201310

utkarshsaxena.2305@gmail.com

3. SHIVAM TIWARI B.Tech
IIMT COLLEGE OF
ENGINEERING GREATER
NOIDA ,201310

tiwarishivam1689@gmail.com

4. ANUKUL RAJ B.Tech
IIMT COLLEGE OF
ENGINEERING GREATER
NOIDA ,201310

anukulrashu@gmail.com

ASHISH SHRIVASTAVA (Mentor) ,

ashish4397_gn@iimtindia.net

Abstract—In today's fast-paced and dynamic world, managing your finances has never been more critical. Whether you're an individual striving to achieve your personal financial goals or a business owner navigating the complexities of financial management, FINOVA is here to simplify the way you handle your finances.

With FINOVA, you can take charge of your financial future, gain a clear understanding of your financial situation, and make informed decisions to achieve your financial goals. Say goodbye to financial stress and uncertainty and welcome a brighter, financially secure tomorrow with FINOVA.

I. INTRODUCTION

FINOVA is a sophisticated financial management solution created to give people and companies effective financial management resources. FINOVA provides a complete solution for long-term financial planning, investment research, tracking spending, and budgeting in the dynamic and more digital financial landscape of today. The main characteristics and advantages of FINOVA are outlined in this abstract.

By streamlining financial management, FINOVA helps people and organizations take charge of their money, feel less stressed about it, and achieve their financial objectives. FINOVA is a useful tool for everyone trying to achieve stability and success in their finances because of its comprehensive features, user-friendly interface, and data security.

FINOVA is an all-inclusive finance management tool created to provide people, companies, and groups effective financial control and decision-making abilities. FINOVA is a feature-rich, intuitive platform that simplifies financial processes, promotes transparency, and enhances financial performance in an increasingly complicated financial environment.

- Entire Financial Situation Overview: FINOVA's easily customizable financial dashboard provides you with a comprehensive picture of your entire financial status. Credit cards, bank accounts, investments, loans, and other things are covered here.
- It's easy to monitor your daily spending, categorize transactions, and analyze your spending habits with expense monitoring to identify areas for improvement and cost savings.
- Budgeting Made Simple: Easily develop, manage, and monitor your spending plan. You may set financial goals, track your progress, and make last-minute adjustments with FINOVA's user-friendly interface.
- Investment Portfolio Management: With FINOVA, investors can keep an eye on their assets, assess their performance, and make informed decisions to grow their wealth.
- Notes Regarding Bill Payment: Never forget to make a payment again. By alerting you to upcoming invoices, FINOVA will help you remain on top of your financial obligations.
- Financial Reports and Insights: With the help of thorough reports and analytics, you may learn a great deal about your financial habits, which will help you make wise decisions and changes.

Financial Reports and Insights: With the help of thorough reports and analytics, you may learn a great deal about your

- financial habits, which will help you make wise decisions and changes.

Notes Regarding Bill Payment: Never forget to make a payment again. By alerting you to upcoming invoices, FINOVA will help you remain on top of your financial obligations. **Some related projects**

Whether it is for small enterprises, personal finance, or major corporations, there are many project apps and software solutions accessible. These are a few well-liked programs for finance management projects:

- **QuickBooks:** For small enterprises, QuickBooks is a feature-rich accounting and financial management program. It has functions for payroll, cost monitoring, invoicing, and more.
- **Mint:** Mint is a free program for managing personal finances that lets users keep track of their earnings, outlays, and investments in one location. It also has tools for goal-setting and budgeting.
- **YNAB, or "You Need a Budget,"** is an app for creating budgets that assists users in setting priorities for their spending, allocating their income to different categories, and setting aside money for future goals.
- **Quicken:** This personal finance program includes capabilities for managing bills, tracking investments, budgeting, and financial planning.
- **Personal Capital:** This financial dashboard assists users in tracking their net worth, managing their investments, and making retirement plans
- **Tiller:** Tiller is a spreadsheet-based financial management program that refreshes your Google Sheets or Excel financial data automatically and connects to your bank accounts.
- **Wave:** For small firms, Wave offers free accounting and finance software. It provides functions including payroll, spending tracking, and invoicing.
- **Online accounting software for small businesses** is called Zoho Books. It has tools for financial reporting, managing expenses, and creating invoices.

SAP Business One is a small and medium-sized business-focused enterprise resource planning (ERP)

solution. It provides CRM, inventory, and financial management features.

- **Oracle NetSuite:** For larger enterprises, NetSuite is a cloud-based ERP solution. On a single platform, it offers e-commerce, CRM, financial administration, and other business operations.
- **Xero:** Xero is an online accounting program with capabilities for bank reconciliation, expense tracking, and invoicing small businesses.
- **FreshBooks:** Designed for small businesses and independent contractors, FreshBooks is a cloud-based accounting program. It has functions for tracking expenses, time, and invoicing.
- **Sage Intacct:** For medium-sized to big enterprises, Sage Intacct is a cloud-based financial management solution. It offers analytics, reporting, and financial consolidation solutions.
- **Mvelopes:** Mvelopes is a personal finance management tool that helps users limit their spending by utilizing the envelope budgeting method.
- **Moneydance:** This personal finance program has tools for bill payment, investment tracking, and budgeting.
- **GnuCash:** For use by individuals and small businesses, GnuCash is free and open-source accounting software. It provides financial reporting, budgeting, and double-entry bookkeeping.

Features of FINOVA

This summary's main focus is on examining the things that FINOVA has to offer. We will carefully examine each feature of the application here. Among the subjects discussed are:

1. **User-Friendly Interface:** An interface that is easy to use and intuitive, with the goal of offering a seamless experience.
2. **Account Aggregation:** FINOVA provides a comprehensive picture of an individual's financial health by combining financial data from several sources.
3. **Expense Tracking:** Comprehensive guides to keeping track of, classifying, and budgeting for daily expenses.
4. **Payment:** A tool that makes it simple for users to handle regular payments and pay their bills.
5. **Investment Tracking:** How FINOVA makes real-time investment tracking and analysis easier.

6. Savings and Goal Planning: This section of the application provides tools for tracking progress and defining financial objectives.

7. Security Measures: Detailed explanation of the strong security measures that protect users' financial information.

The Rise of FINOVA

We will now trace the origins and development of FINOVA. We'll examine the application's motivation, the development process, and the decision to use Flutter technology as the platform to create this financial management tool.

Understanding Flutter

Understanding the technology behind FINOVA is crucial before delving into its intricacies. The open-source Flutter framework, created by Google, is a flexible and dynamic tool for creating cross-platform applications. This chapter will give a thorough explanation of Flutter, its salient characteristics, and the rationale behind choosing it as the FINOVA framework.

User Experience and Design

Any mobile application's user experience and design are critical to its success. We'll talk about the user-centric development methodology used for FINOVA in this part. Subjects covered include the iterative development process, user feedback, and UI/UX design principles.

Design and user experience (UX) are essential components of any finance management software, such as FINOVA. A well-designed software guarantees that users may effectively manage their finances while having fun, in addition to improving usability. Key factors for design and UX in FINOVA are as follows:

1. First, user-centered design

User Research: To fully grasp the requirements and problems of your intended audience, conduct in-depth user research. The design choices for the app should be influenced by this research.

User Personas: To represent various user base groups, create user personas. This will make it easier to modify the program to fit the unique needs of various users.

2. Natural Guidance:

- **Simplified Menus:** Maintain an easy-to-use and basic navigation menu system. To help users navigate the app, utilize labels and icons that are easy to grasp.

- **Info Hierarchy:** Organize financial data in a hierarchical manner, making the most crucial and often utilized features readily available.

3. A Clear and Uniform User Interface:

- **User Interface (UI) Consistency:** Keep the app's design language constant. To establish a unified visual identity, use a harmonized color scheme, font, and layout.

- **Whitespace:** Make good use of whitespace to clear up clutter and give the program a tidy, orderly appearance.

4. Adherence to Mobile Devices:

- **Responsive Design:** Make sure FINOVA is completely responsive to offer an excellent experience across a range of screen sizes and gadgets, including tablets and smartphones.

- **Touch-Friendly Controls:** Provide mobile users with touch-friendly input fields and buttons.

5. Availability:

- **Accessibility Features:** Add accessibility features so that users with disabilities can utilize the application. Support for high contrast modes, keyboard navigation, and screen readers are all included in this.

6. Trust and Security:

- **Safe Data Handling:** Reassure users of the security aspects of the application, such as two-factor authentication, encryption, and safe data storage.

7. Individualization

- **Customization:** Give customers the option to personalize the app by letting them choose a theme, arrange dashboard widgets, and establish financial objectives.

- **Suggestions:** Using user data, provide tailored financial advice and insights.

8. Suggestions and Assistance:

- **In-App Help:** Provide lessons and in-app help for users who might require assistance.

- **Feedback Mechanisms:** To handle user inquiries and concerns, implement chat support or feedback forms.

9. Execution:

- **Speed and Responsiveness:** Make the application as quick and responsive as possible. Users may become irritated by slow loading times.

10. Inaccessible Data:

- **Offline Mode:** Provides consumers with the ability to view vital financial data even when they are not online. When the connection is reestablished, data synchronization ought to take place.

11. Visualization of Data:

- Graphs and Charts: Use interactive graphs and charts to visualize financial data and provide users a quick overview of their financial situation.

12. Evaluate and Retest:

- Usability Testing: To find problems and potential areas for improvement, test the app frequently with actual users.

- Iterative Design: Apply user feedback and continuously enhance the functionality and appearance of the app.

13. Compatibility Across Platforms:

- Multi-Platform Design: Take into account creating FINOVA for both iOS and Android to provide a unified user experience.

14. Beauty:

- Appealing Visuals: Use eye-catching design cues and captivating animations to make the app visually appealing to consumers.

Benefits and Advantages of FINOVA

Adopting FINOVA is expected to yield numerous advantages for both individuals and enterprises. These benefits—which include financial knowledge, time savings, and more effective decision-making—are examined in this chapter.

For both people and companies, the FINOVA money managing mobile application has a number of advantages. The following are some of the main benefits of utilizing FINOVA:

1. Simplified Financial Management: FINOVA offers a unified platform that makes budgeting, cost tracking, and financial planning easier for individuals or businesses.

2. Real-Time Access: Users can get real-time insights into their financial transactions and overall health at any time and from any location by accessing their financial data.

3. Expense Tracking: Using FINOVA, users may keep tabs on and organize their spending, assisting both individuals and companies in spotting trends in their spending and possible places to save costs.

4. Financial Planning and Budgeting: - The app helps users save money for the future and keep track of their spending by letting them create and monitor budgets.

5. Investment Management: By connecting investment accounts to FINOVA, users may track investments, keep an eye on portfolio performance, and more easily plan for retirement or other financial objectives.

6. Reminders for Bill Payments: FINOVA can send out reminders for bill payments and deadlines, which lowers the possibility of overdue payments and penalties.

7. Security and Privacy: To protect sensitive financial information, the app uses strong security mechanisms like authentication and encryption.

8. Data Visualization: FINOVA provides a visual representation of financial data through interactive charts and graphs, which helps users better comprehend their financial situation.

9. Financial Insights: Using user data, the app may offer individualized financial insights and recommendations to assist users in making wise financial decisions.

10. User-Friendly Interface: FINOVA has an easy-to-use interface that makes it possible for both new and expert users to operate the app with ease.

11. Cross-Platform Compatibility: FINOVA is accessible and flexible to a broad spectrum of consumers, being available on both the Android and iOS platforms.

12. Offline Access: Users can still access vital financial data even when they are not online. Reestablishing an internet connection triggers synchronization.

13. Less Paperwork: FINOVA makes financial records digital, which lessens the need for physical paperwork and increases efficiency and the environment.

14. Customization: - By selecting themes, allocating dashboard widgets, and establishing their financial objectives, users may make their FINOVA experience uniquely their own.

15. Transparent Reporting: - To evaluate their financial performance or for tax purposes, users can create comprehensive financial reports and statements.

16. Financial Goal Tracking: Whether a user wants to pay off debt, buy a house, or save for a vacation, FINOVA can help them plan, monitor, and accomplish their financial objectives.

17. Security notifications: - To improve financial security, users can sign up to receive notifications when transactions seem suspicious or perhaps fraudulent.

Conclusion

In the last chapter, we restate the synopsis's main ideas while highlighting FINOVA's significance for contemporary finance management. For readers who are interested in app development and financial technology, we also offer insights and takeaways.

To sum up, FINOVA is a strong and user-friendly mobile application for managing finances that has many advantages for both individuals and companies. We have examined the many benefits and features that make FINOVA an invaluable instrument for financial management in this summary.

By utilizing the flexible Flutter framework, FINOVA's development and implementation were able to deliver a smooth and responsive user experience on a variety of platforms. The user was the primary designer of the application, which guarantees easy navigation, a streamlined and uniform interface, accessibility, and security features.

Simplified financial management, instant access to financial information, tools for financial planning and budgeting, tracking of expenses, and investment management are all advantages of using FINOVA. In order to make sure customers never forget significant deadlines, the software also enables bill payment reminders. Secure financial data is protected thanks to strong security protocols and a dedication to data privacy.

References

Building a finance management mobile application, such as FINOVA, requires you to use a variety of tools to guide your development endeavors. The following references and sources might offer insightful analysis and helpful direction:

1. Books:

18. Usability and Accessibility: - All users, including those with disabilities, will have a flawless experience with the app because it was developed with usability and accessibility in mind.

19. Time Savings: - FINOVA saves users time that would be used for manual computations and record-keeping by automating financial activities.

20. Customer Support: To answer user queries and issues, the app may provide in-app help, customer support, and feedback options.

All things considered, FINOVA makes financial administration easier, improves financial literacy, and aids users in gaining more security and control over their money. It provides a complete solution for maintaining and enhancing financial well-being for both individuals and enterprises.

a. Theresa Neil's book "Mobile Design Pattern Gallery" offers best practices and design patterns for mobile app interfaces, which are applicable to the UI of FINOVA.

b. Luke Wroblewski's "Mobile First" focuses on developing mobile-optimized user experiences and approaches to mobile app design.

2. Research papers and academic journals:

a. Studies on the security, usability, and user experience of mobile apps can provide information about recommended practices and potential traps to be aware of.

3. Online Tutorials & Courses:

a. You can find classes on finance and accounting as well as mobile app development and design using Flutter-related technologies on websites like Coursera, Udemy, edX, and Pluralsight.

4. Blogs for Mobile App Development and Design:

a. Articles and case studies on mobile app design and development may be found on blogs like Awwwards, UX Design, and Smashing Magazine.

-

FINOVA can provide a visually appealing and easy-to-use experience by carefully considering these UX and design factors, which elevates it to the status of a reliable and efficient financial management tool. Keep in mind that design and user experience should be continuous processes that adjust to changing technological developments and human input.

5. Flutter Records:

a. Detailed instructions for utilizing the Flutter framework to construct mobile apps may be found in the official Flutter documentation (<https://flutter.dev/docs>).

6. Accounting and Finance Resources:

a. Sources of information on financial management and planning include accounting books, financial websites, and articles from publications like the Financial Times, The Balance, and Investopedia.

7. Resources for UX and UI Design:

a. Articles, courses, and research on UX and UI design principles can be found at UX design resources such as Nielsen Norman Group (www.nngroup.com) and Interaction Design Foundation (www.interaction-design.org).

8. User Input and Case Studies:

a. Examine case studies of profitable money management apps and get user input to learn what functions well and poorly in this field.

9. Reviews of the Play Store and App Store:

a. User expectations and potential development areas can be discerned from reviews and ratings of current finance management apps.

Coding

```
import 'package:flutter/material.dart';
import 'package:flutter_svg/flutter_svg.dart';
import 'package:flutter_wallet/util/file_path.dart';

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Theme.of(context).backgroundColor,
      body: Padding(
        padding: const EdgeInsets.only(left: 18, right: 18, top: 34),
        child: SingleChildScrollView(
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: <Widget>[
              _contentHeader(),
              const SizedBox(
                height: 30,
              ),
              Text(
                'Account Overview',
                style: Theme.of(context).textTheme.headline4,
              ),
              const SizedBox(
                height: 16,
              ),
              _contentOverView(),
              const SizedBox(
                height: 30,
              ),
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: <Widget>[
                  Text(
                    'Send Money',
                    style: Theme.of(context).textTheme.headline4,
                  ),
                  SvgPicture.asset(
                    scan,
                    color: Theme.of(context).iconTheme.color,
                    width: 18,
                  ),
                ],
              ),
              const SizedBox(height: 16),
              _contentSendMoney(),
              const SizedBox(
                height: 30,
              ),
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: <Widget>[
                  Text(
                    'Services',
                    style: Theme.of(context).textTheme.headline4,
                  ),
```

```
SvgPicture.asset(  
  filter,  
  color: Theme.of(context).iconTheme.color,  
  width: 18,  
),  
],  
),  
const SizedBox(  
  height: 16,  
),
```

Coding

```
_contentServices(context),
  ],
),
),
),
);
}

Widget _contentHeader() {
return Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: <Widget>[
    Row(
      children: <Widget>[
        SvgPicture.asset(
          logo,
          width: 34,
        ),
        const SizedBox(
          width: 12,
        ),
        Text(
          'eWalle',
          style: Theme.of(context).textTheme.headline3,
        )
      ],
    ),
    InkWell(
      onTap: () {
        setState(() {
          // print('call');
          // xOffset = 240;
          // yOffset = 180;
          // scaleFactor = 0.7;
          // isDrawerOpen = true;
        });
      },
      child: SvgPicture.asset(
        menu,
        width: 16,
        color: Theme.of(context).iconTheme.color,
      ),
    ),
  ],
);
}

Widget _contentOverView() {
return Container(
  padding: const EdgeInsets.only(left: 18, right: 18, top: 22, bottom: 22),
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(10),
    color: Theme.of(context).cardColor,
    // color: const Color(0xffF1F3F6),
  ),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: <Widget>[
      Column(

        crossAxisAlignment: CrossAxisAlignment.start,
```

children:

```
<Widget>[ Text(  
  '20,600',  
  style: Theme.of(context).textTheme.headline5,  
)  
const SizedBox( height:  
  12,  
)  
Text(  
  'Current Balance',  
  style: Theme.of(context).textTheme.headline4!.copyWith( fontSize: 15,  
    fontWeight:  
    FontWeight.w400, ),
```

Coding

```
)
  ],
),
Container(
  height: 55,
  width: 55,
  decoration: BoxDecoration(
    color: const Color(0xffffac30),
    borderRadius: BorderRadius.circular(80),
  ),
  child: const Center(
    child: Icon(
      Icons.add,
    ),
  ),
),
),
],
),
);

}

Widget _contentSendMoney() {
return SizedBox(
  height: 100,
  child: ListView(
    scrollDirection: Axis.horizontal,
    children: <Widget>[
      Container(
        width: 80,
        padding: const EdgeInsets.only(
          left: 18,
          right: 18,
          top: 28,
          bottom: 28,
        ),
        child: Container(
          height: 10,
          width: 10,
          decoration: const BoxDecoration(
            color: Color(0xffffac30),
            shape: BoxShape.circle,
          ),
          child: const Center(
            child: Icon(
              Icons.add,
            ),
          ),
        ),
      ),
      Container(
        margin: const EdgeInsets.only(right: 10),
        padding: const EdgeInsets.all(16),
        width: 80,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(10),
          color: Theme.of(context).cardColor,
        ),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: <Widget>[
            Container(
              decoration: BoxDecoration(
```

```
shape: BoxShape.circle,  
color: Colors.white,  
border: Border.all(color: const Color(0xffD8D9E4)),  
child: CircleAvatar(  
  radius: 22.0,  
  backgroundColor: Theme.of(context).backgroundColor,  
  child: ClipRRect(  
    child: SvgPicture.asset(avatorOne),  
    borderRadius: BorderRadius.circular(50.0),  
  ),  
),  
),
```

```
),
    Text(
        'Joseph',
        style: Theme.of(context).textTheme.bodyText1,
    )
],
),
),
Container(
    margin: const EdgeInsets.only(right: 10),
    padding: const EdgeInsets.all(16),
    width: 80,
    decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(10),
        color: Theme.of(context).cardColor,
    ),
    child: Column(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: <Widget>[
            Container(
                decoration: BoxDecoration(
                    shape: BoxShape.circle,
                    border: Border.all(color: const Color(0xffD8D9E4)),
                ),
                child: CircleAvatar(
                    radius: 22.0,
                    backgroundColor: Theme.of(context).backgroundColor,
                    child: ClipRRect(
                        child: SvgPicture.asset(avatorThree),
                        borderRadius: BorderRadius.circular(50.0),
                    ),
                ),
            ),
            Text(
                'Ashley',
                style: Theme.of(context).textTheme.bodyText1,
            )
        ],
    ),
);
}
Widget _contentServices(BuildContext context) {

    List<ModelServices> listServices = [];

    listServices.add(ModelServices(title: "Send\nMoney", img: send));
    listServices.add(ModelServices(title: "Receive\nMoney", img:
    recive)); listServices.add(ModelServices(title: "Mobile\nPrepaid",
    img: mobile)); listServices
        .add(ModelServices(title: "Electricity\nBill", img: electricity));

    listServices.add(ModelServices(title: "Cashback\nOffer", img: cashback));
    listServices.add(ModelServices(title: "Movie\nTickets", img: movie));
    listServices.add(ModelServices(title: "Flight\nTickets", img: flight));
    listServices.add(ModelServices(title: "More\nOptions", img: menu));

    return SizedBox(
        width: double.infinity,
        height: 400,
```

```
child: GridView.count(  
  crossAxisCount: 4,  
  childAspectRatio: MediaQuery.of(context).size.width /  
    (MediaQuery.of(context).size.height / 1.1),  
  children: listServices.map((value) {  
    return GestureDetector(  
      onTap: () {  
        // print('${value.title}');  
      },
```

```
),
  Text(
    'Joseph',
    style: Theme.of(context).textTheme.bodyText1,
  )
],
),
),
Container(
  margin: const EdgeInsets.only(right: 10),
  padding: const EdgeInsets.all(16),
  width: 80,
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(10),
    color: Theme.of(context).cardColor,
  ),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: <Widget>[
      Container(
        decoration: BoxDecoration(
          shape: BoxShape.circle,
          border: Border.all(color: const Color(0xffD8D9E4)),
        ),
        child: CircleAvatar(
          radius: 22.0,
          backgroundColor: Theme.of(context).backgroundColor,
          child: ClipRRect(
            child: SvgPicture.asset(avatorThree),
            borderRadius: BorderRadius.circular(50.0),
          ),
        ),
      ),
      Text(
        'Ashley',
        style: Theme.of(context).textTheme.bodyText1,
      )
    ],
  ),
),
);
}
Widget _contentServices(BuildContext context) {

  List<ModelServices> listServices = [];

  listServices.add(ModelServices(title: "Send\nMoney", img: send));
  listServices.add(ModelServices(title: "Receive\nMoney", img:
  recive)); listServices.add(ModelServices(title: "Mobile\nPrepaid",
  img: mobile)); listServices
  .add(ModelServices(title: "Electricity\nBill", img: electricity));

  listServices.add(ModelServices(title: "Cashback\nOffer", img: cashback));
  listServices.add(ModelServices(title: "Movie\nTickets", img: movie));
  listServices.add(ModelServices(title: "Flight\nTickets", img: flight));
  listServices.add(ModelServices(title: "More\nOptions", img: menu));

  return SizedBox(
    width: double.infinity,
    height: 400,
```

```
child: GridView.count(  
  crossAxisCount: 4,  
  childAspectRatio: MediaQuery.of(context).size.width /  
    (MediaQuery.of(context).size.height / 1.1),  
  children: listServices.map((value) {  
    return GestureDetector(  
      onTap: () {  
        // print('${value.title}');  
      },
```

```
) child: Column(  
  crossAxisAlignment: CrossAxisAlignment.center,  
  children: <Widget>[  
    Container(  
      width: 50,  
      height: 50,  
      padding: const EdgeInsets.all(12),  
      decoration: BoxDecoration(  
        borderRadius: BorderRadius.circular(8),  
        color: Theme.of(context).cardColor,  
      ),  
      child: SvgPicture.asset(  
        value.img,  
        color: Theme.of(context).iconTheme.color,  
      ),  
    ),  
    const SizedBox(  
      height: 8,  
    ),  
    Text(  
      value.title,  
      textAlign: TextAlign.center,  
  
      style: Theme.of(context).textTheme.bodyText1,  
    ),  
    const SizedBox(  
      height: 14,  
    ),  
  ],  
),  
);  
}).toList(),  
),  
);  
}  
  
}  
  
class ModelServices {  
  String title, img;  
  ModelServices({required this.title, required this.img});  
  
}
```

Output

