**SYNOPSIS**

A job portal, also known as an online job board or job search engine, is a website that allows employers to post job vacancies and job seekers to search and apply for jobs. The portal acts as an intermediary between job seekers and employers, providing a platform for them to connect and communicate.

Job portals typically offer a range of features such as job search filters, resume uploading, application tracking, and email alerts. Job seekers can create profiles on the portal, which may include their resumes, cover letter, and other relevant information. Employers can search these profiles to find suitable candidates for their job openings.

Job portals can be industry-specific, catering to a particular sector such as IT, healthcare, or finance. They can also be location-specific, targeting a particular city or region. Some job portals are global in scope and offer job listings from all around the world.

In summary, a job portal is an online platform that connects job seekers with potential employers, providing them with a range of tools and resources to facilitate the job search process.

Employers can use the app to post job openings, review resumes, and schedule interviews with potential candidates. The app may also offer additional features, such as job alerts, salary information, and career advice.

The job portal app can be accessed through multiple devices, including mobile devices and desktop computers, making it convenient for job seekers to stay connected with potential employers.

With its wide range of features and accessibility, a job portal app can make the job search process more efficient and convenient for both job seekers and employers.

**1.INTRODUCTION**

In today's digital age, job portals have become an indispensable tool for both job seekers and employers. These online platforms offer a centralized location for job seekers to find job vacancies and for employers to post job openings, making the job search process more efficient and convenient for both parties.

Job portals typically offer a range of features such as job alerts, resume uploading, and application tracking, which make it easier for job seekers to find and apply for suitable job vacancies. Employers, on the other hand, can search through the database of job seekers' profiles to find the most qualified candidates for their job openings.

Job portals can be industry-specific, location-specific, or global in scope, providing a variety of options for job seekers and employers. They also offer additional resources such as career advice, interview tips, and salary information, making them a valuable tool for anyone looking to advance their career.

Overall, job portals have transformed the job search process by providing a more streamlined and efficient way for job seekers and employers to connect and communicate. With the abundance of options available, job portals have become an essential tool for anyone looking to find their dream job or the perfect candidate for their organization.

**1.1SYSTEM SPECIFICATION**

**1.1.1HARDWARE SPECIFICATION**

Processor : Ryzen 5

Speed : 2.10GHZ

RAM Capacity : 8 GB

Hard Disk Drive : 474GB

Keyboard : 108 Keys

Mouse : Lenovo

Mother Board : AMD

Screen : 15 inch

**1.1.2 SOFTWARE SPECIFICATION**

Operating system : Windows 11

Front End : Dart

Style and Script : Kotlin, java

**1.1.3SOFTWARE FEATURES**

**Intelligent code editor**: Android Studio has a powerful code editor that provides advanced code completion, syntax highlighting, and refactoring tools. It also supports multiple programming languages such as Java, Kotlin, and C++.

**Layout editor**: The layout editor in Android Studio allows developers to easily create and design complex user interfaces (UIs) using drag-and-drop tools and visual editors.

**Gradle build system**: Android Studio uses the Gradle build system, which is a flexible and efficient system that automates the build process and supports customizable build configurations.

**Emulator**: Android Studio comes with a built-in emulator that enables developers to test their apps on virtual devices that emulate different Android devices and operating system versions.

**Debugger:** The debugger in Android Studio allows developers to easily identify and fix issues in their code during development.

**Android Virtual Device (AVD) Manager**: The AVD manager in Android Studio allows developers to create and manage virtual devices for testing their apps on various Android device configurations.

**Code templates:** Android Studio provides pre-built code templates that allow developers to quickly create common code patterns such as menus, dialogs, and settings screens.

**Widget library**: Flutter provides a rich set of customizable widgets that can be easily combined to create beautiful and responsive UIs for mobile, web, and desktop platforms.

**Hot Reload**: Flutter's Hot Reload feature allows developers to instantly see the changes they make to the code in their app, without needing to recompile or restart the app.

**Dart programming language**: Flutter uses the Dart programming language, which is an object-oriented language that compiles to native code and offers features such as garbage collection and just-in-time compilation.

**Platform-specific APIs**: Flutter allows developers to access platform-specific APIs for integrating with native features of the device, such as camera, sensors, and location services.

**Material Design and Cupertino widgets**: Flutter includes pre-built Material Design and Cupertino widgets, which provide a consistent look and feel across Android and iOS platforms respectively.

**Stateful Hot Reload:** Flutter's Stateful Hot Reload feature allows developers to make changes to the app's state and see the effects of those changes immediately without losing the current app state.

**Testing tools:** Flutter comes with built-in testing tools, such as unit testing, widget testing, and integration testing, which enable developers to write tests and ensure the app's functionality is working as expected.

**Development tools**: Flutter provides development tools such as DevTools, which allows developers to debug and profile their apps, and Flutter Inspector, which provides a visual representation of the app's widget tree.

**Customizable themes**: Flutter allows developers to customize the look and feel of their app using themes, which can be easily applied to the app's widgets.

**Accessibility:** Flutter provides support for accessibility, such as screen readers and text-to-speech, to ensure that the app can be used by everyone, including people with disabilities.

2. **SYSTEM STUDY**

**2.1EXISTING SYSTEM**

**Job Search:** Job seekers can search for job vacancies by keywords, location, salary range, industry, job type, and other criteria. Job search results can be displayed with relevant job details such as job title, job description, salary, location, and application deadline.

**Resume/CV Builder**: Job seekers can create their resumes/CVs online, which can be used to apply for job vacancies through the job portal. The resume builder typically includes fields for personal information, work experience, education, skills, and other relevant details.

**Job Application:** Job seekers can apply for job vacancies through the job portal by uploading their resumes/CVs, cover letters, and other supporting documents. Employers can receive and review job applications through the job portal, and contact the applicants through the portal or via email.

**2.1.1DISADVANTAGES**

**Competition:** Due to the ease of access to job vacancies through job portals, there can be a high level of competition for job vacancies, making it more challenging for job seekers to stand out from the crowd.

**Limited personal interaction:** The lack of personal interaction between job seekers and employers can make it difficult for employers to assess a candidate's personality, work ethic, and cultural fit. Similarly, job seekers may not have the opportunity to ask questions or learn more about the company before applying.

**2.2PROPOSED SYSTEM**

A proposed job portal system could offer several features to enhance the user experience for both job seekers and employers. One such feature could be advanced search filters, which would allow job seekers to refine their job search based on specific criteria such as job type, location, salary range, industry, and experience level.

Employers could create detailed company profiles that provide information about their mission, values, culture, benefits, and employee testimonials to help job seekers make more informed decisions.

An AI-powered candidate matching system could help employers find suitable candidates by analyzing factors such as skills, experience, and cultural fit. Job seekers could create video resumes to showcase their skills and personality, and take skills assessments to demonstrate their proficiency in specific areas. Virtual interviews using video conferencing software could make the hiring process more accessible and convenient for both employers and job seekers.

A chatbot assistant could provide support and answer common questions throughout the hiring process, while a mobile app would provide greater flexibility and convenience for users.

The platform could also provide salary data for different job types and industries, and allow job seekers and employers to leave reviews and ratings for each other, promoting transparency and accountability in the hiring process.

2.2.1 FEATURES OF PROPOSED SYSTEM

A proposed job portal system could include a range of features to improve the user experience for both job seekers and employers.

Advanced search filters would allow job seekers to refine their job search by factors such as job type, location, salary range, industry, and experience level. Employers could create detailed company profiles that include information about their mission, values, culture, benefits, and employee testimonials.

Employers could use an AI-powered candidate matching system to find suitable candidates for their job openings based on factors such as skills, experience, and cultural fit. Job seekers could create video resumes to showcase their skills and personality, while taking skills assessments to validate their proficiency in specific areas.

Virtual interviews using video conferencing software would make the hiring process more convenient and accessible.

Salary data would be provided for different job types and industries, allowing job seekers to make informed decisions about their career paths.

The platform could also provide job seekers with resources such as career advice, job market trends, and networking opportunities.

**3.1SYSTEM DESIGN AND DEVELOPMENT**

**3.1FILE DESIGN**

main. dart is the main entry point of a Flutter application written in Dart programming language. It is a file that contains the code for the main function, which is the starting point of the application. In this file, you can declare and initialize variables, define functions, and create the structure of your application.

The main function is responsible for initializing the application and setting up the root widget of the widget tree. This widget tree is composed of a hierarchy of widgets that define the layout and behavior of the application. The runApp() method is used to start the application and pass in the root widget as a parameter.

In main. dart, you can also define routes for different screens of your application using the MaterialApp widget. This allows you to navigate between different screens or pages of the application using the Navigator widget.

Overall, the main. dart is a crucial file in a Flutter application as it sets up the fundamental structure of the app and provides a starting point for the program execution.

**3.2INPUT DESIGN**

Designing a job portal app input form requires careful consideration to ensure that it is user-friendly and efficient. One important aspect of this is to keep the form simple and easy to navigate, avoiding clutter and unnecessary information. Clear labels should be used for each field to help users understand what information is required and to reduce errors. It's also important to ensure that the form is accessible to all users, including those with disabilities. Using appropriate colors and fonts, and ensuring that the form can be navigated using a keyboard, can help achieve this.

Providing helpul tips and instructions where necessary can guide users through the input process. Autocomplete functionality can also be useful to help users enter information more quickly and accurately, while dropdown menus can help users select options more easily. Real-time input validation is crucial to provide feedback to users about errors or missing information.

Allowing users to save their progress and return to the input form later can improve the user experience, as can testing the form with real users to ensure that it is easy to use and understand. By following these guidelines, you can design a job portal app input form that is efficient, user-friendly, and accessible to all.

**3.3OUTPUT DESIGN**

The output design for a job portal app is crucial in providing users with the information they need to make informed decisions about job opportunities. To ensure that the output is effective, it should be simple, easy to read, and understandable. Clear and concise language should be used, avoiding jargon that may be unfamiliar to some users. The information should be organized effectively, making it easy for users to navigate and find what they need. Headings, bullet points, and lists can help to make the information easier to read.

Visual elements such as charts, graphs, and images can be used to make the output more engaging and easier to understand. This can help users to quickly identify key information and make informed decisions about job opportunities. Additionally, providing users with the ability to customize the output can improve the user experience, allowing them to tailor the output to their specific needs and preferences.Real-time updates can also be helpful, providing users with up-to-date information on job opportunities as they become available. Finally, it's important to test the output design with real users to ensure that it is effective and meets their needs. By following these guidelines, you can design an effective output for your job portal app that provides users with the information they need to make informed decisions about job opportunities.

**3.5 SYSTEM DEVELOPMENT**

**3.5.1 DESCRIPTION OF MODULES**

There are 3 modules in the project. They are as follows

* homepage Module
* bottom\_menu\_bar Module
* job\_detail\_pageModule

**homepage Module:** The home. dart file might contain code related to defining the layout and behavior of this primary screen, such as defining the structure of the UI and handling user input. It could also contain code related to fetching or displaying data relevant to the home screen, such as recent activity or recommended content.

**bottom\_menu\_bar Module:** The bottom\_menu\_bar. Dart file might contain code related to defining the appearance and behavior of this navigation bar, such as setting the layout, styling, and animations. It might also contain code related to handling user input and navigating to different parts of the application when a user selects an icon or label on the bottom bar.

**job\_detail\_pageModule:** The job\_detail\_page.dart file might contain code related to defining the appearance and behavior of the job detail page, such as setting up the UI elements, styling, and animations. It might also contain code related to fetching and displaying information about the job listing, such as the job title, description, requirements, and application process.

**4.TESTING AND IMPLEMENTATION**

The common view of testing held by users is that it is performed to prove that there are no errors in a program. This is extremely difficult since designer cannot prove to be one hundred percent accurate. Therefore, the most useful and practical approach is with the understanding that testing is the process of executing a program with explicit intention of finding errors that make the program fail.

Testing has its own cycle. The testing process begins with the product requirements phase and from there parallels the entire development process. In other words, for each phase of the development process there is an important testing activity. Successful testing requires a methodical approach. It requires focusing on basic critical factors:

* Planning
* Project and process control
* Risk management
* Inspections
* Measurement tools
* Organization and professionalism

**4.1TEST PLAN**

Before going for testing, first we have to decide upon the type of testing to be carried out. The following factors are taken into consideration:

* To ensure that information properly flows into and out of program
* To find out whether the local data structures maintains its integrity during all steps in an algorithm execution
* To ensure that the module operate properly at boundaries established to limit or restrict processing
* To find out whether error - handling paths are working correctly or not
* To find out whether the values are correctly updated or not
* Check for validations

**UNIT TESTING**

Unit or module testing is the process of testing the individual components (subprograms or procedures) of a program. The purpose is to discover discrepancies between the modules interface specification and its actual behavior. In our system each module must be tested independently for validation.

**INTEGRATION TESTING**

Integration testing is the process of combining and testing multiple components together. The primary objective of integration testing is to discover errors in the interfaces between the components. In our system each of the modules mentioned above, are tested for checking the integration between them, after each of them are tested individually.

**4.2SYSTEM IMPLEMENTATION**

System implementation is the important stage of project when the theoretical design is tunes into practical system. The main stages in the implementation are as follows:

* Planning
* Training
* System testing and
* Changeover planning

Planning is the first task in the system implementation. Planning is deciding on the method and the time scale to be adapted. At the time of implementation of any system people from different departments and system analysis involve. They are confirmed to practical problem of controlling various activities of people outside their own data processing departments. The line manager controlled through an implementation co-ordinate committee. The committee consists of ideas, Problems and complaints of user department. It must also consider,

* The implementation of system environment.
* Self selection and allocation for implementation tasks.
* Consultation with unions and resources available.
* Standby facilities and channels of communication.

**5.CONCLUSION**

In conclusion, a job portal is a valuable resource for both job seekers and employers. It provides a platform for job seekers to search and apply for relevant job openings, as well as access resources to help them in their job search. Employers can use job portals to post job vacancies and search for potential candidates.Job portals have significantly transformed the traditional job search process, making it more convenient and accessible. With the increasing use of technology, job portals are constantly evolving, providing new features and tools to enhance the user experience.

However, despite the many benefits of job portals, it is important to note that they are not a silver bullet for finding a job. Job seekers should still invest time and effort in networking, building their skills, and tailoring their applications to specific job requirements. Similarly, employers should not rely solely on job portals to find the best candidates and should also consider other recruitment strategies such as employee referrals and social media recruiting.Overall, job portals have transformed the job search and recruitment process, making it more efficient and accessible for everyone involved.

**SCOPE FOR FUTURE ENHANCEMENT:**

Job portals have become essential to the recruitment process, providing a centralized platform for job seekers and employers to connect. While job portals offer a range of features, there is room for future enhancement to improve the user experience and increase efficiency.

One area of potential improvement is the use of artificial intelligence and machine learning algorithms to match job seekers with suitable job openings. By analyzing a candidate's skills, experience, and preferences, job portals can suggest relevant job opportunities, saving time and effort for both job seekers and employers.

Another area for improvement is the integration of social media platforms to enhance networking opportunities. Job portals can allow users to connect their social media profiles, providing employers with a more comprehensive view of a candidate's interests, hobbies, and personality traits. This can also enable job seekers to leverage their existing network to find job opportunities.

**BIBLIOGRAPHY**

**REFERENCE BOOKS**

* Get Hired!: Winning Strategies to Ace the Interview" by Paul C. Green
* "Job Searching with Social Media For Dummies" by Joshua Waldman
* The 2-Hour Job Search: Using Technology to Get the Right Job Faster" by Steve Dalton

**REFERENCE WEBSITES**

* Indeed - https://www.indeed.com/
* Glassdoor **-** https://www.glassdoor.com/
* LinkedIn **-** https://www.linkedin.com/jobs/
* Monster **-** https://www.monster.com/
* CareerBuilder **-** https://www.careerbuilder.com/

**APPENDICES**

**A. DATA FLOW DIAGRAM**

Dataflow diagram is a representation in which overall description of a system can be shown in the form of a diagram. There are four symbols that are used in the design.

* Entities

External entities represent the sources of data that enter the system or the recipients of data that leave the system

* Process

Processes represent activities in which data is manipulated by being stored or retrieved or transformed in some way. A circle represents it. The process will show the data transformation or change.

* Database

Databases represent storage of data within the system

* Data Flow

A data flow shows a flow of information from its source to its destination. A line represents a data flow, with arrow heads showing the direction of flow.

Dataflow diagram is a representation in which overall description of a system can be shown in the form of a diagram. The description of the system will start from the source of the content and with its destination. The diagram can be divided many different stages and each stage must give information about the system. Data Flow Diagram means of representing a system at any level of details with a graphic network of symbols showing data flows, data stores, data processes, and data sources/destination.The dataflow diagram can be divided into two levels. The first stage gives the brief pictorial overall description of the system. This will be containing user input, process and the destination. The source and the destination must be in the square box and the process will be in the circle. This level is shown in below diagram.

**Level 0**

Job name

Role and resposibiltydetails

admin

apply

user

**C. SAMPLE CODING**

import 'package: flutter/material. Dart';

import 'package: flutter\_job\_portal/theme/colors. Dart';

import 'package: flutter\_job\_portal/theme/images. Dart';

import 'package: flutter\_job\_portal/ui/bottom\_menu\_bar.dart';

import 'package: flutter\_job\_portal/ui/job\_detail\_page.dart';

class Homepage extends Stateless Widget {

const Homepage({Key key}) : super(key: key);

Widget \_appBar (BuildContext context) {

return Container(

padding: EdgeInsets.symmetric(horizontal: 16, vertical: 10),

child: Row(

children: [

CircleAvatar(

backgroundImage: AssetImage(Images.user1),

),

Spacer(),

IconButton(

icon: Icon(Icons.notifications\_none\_rounded),

onPressed: () {},

)

],

),

);

}

Widget \_header(BuildContext context) {

return Container(

margin: EdgeInsets.symmetric(vertical: 12),

padding: EdgeInsets.symmetric(horizontal: 16, vertical: 10),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text("Hello Jason",

style: TextStyle(

fontSize: 15,

color: KColors.subtitle,

fontWeight: FontWeight.w500,

)),

SizedBox(

height: 6,

),

Text("Find your perfect job",

style: TextStyle(

fontSize: 20,

color: KColors.title,

fontWeight: FontWeight.bold)),

SizedBox(

height: 10,

),

Row(

children: [

Expanded(

child: Container(

padding: EdgeInsets.symmetric(horizontal: 16, vertical: 12),

decoration: BoxDecoration(

color: KColors.lightGrey,

borderRadius: BorderRadius.circular(5)),

child: Text(

"What are you looking for?",

style: TextStyle(fontSize: 15, color: KColors.subtitle),

),

),

),

SizedBox(

width: 16,

),

Container(

decoration: BoxDecoration(

color: KColors.primary,

borderRadius: BorderRadius.circular(5),

),

height: 40,

child: IconButton(

color: KColors.primary,

icon: Icon(Icons.search, color: Colors.white),

onPressed: () {},

),

)

],

)

],

),

);

}

Widget \_recommendedSection(BuildContext context) {

return Container(

padding: EdgeInsets.symmetric(horizontal: 16, vertical: 10),

margin: EdgeInsets.symmetric(vertical: 12),

height: 200,

width: MediaQuery.of(context).size.width,

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(

"Recommended",

style: TextStyle(fontWeight: FontWeight.bold, color: KColors.title),

),

SizedBox(height: 10),

Expanded(

child: ListView(

scrollDirection: Axis.horizontal,

children: [

\_recommendedJob(context,

company: "Google",

img: Images.google,

title: "UX Designer",

sub: "\$45,000 Remote",

isActive: true),

\_recommendedJob(context,

company: "DropBox",

img: Images.dropbox,

title: "Reserch Assist",

sub: "\$45,000 Remote",

isActive: false)

],

),

),

],

),

);

}

Widget \_recommendedJob(

BuildContext context, {

String img,

String company,

String title,

String sub,

bool isActive = false,

}) {

return Padding(

padding: const EdgeInsets.only(right: 10),

child: GestureDetector(

onTap: () {

Navigator.push(context, JobDetailPage.getJobDetail());

},

child: AspectRatio(

aspectRatio: 1.3,

child: Container(

decoration: BoxDecoration(

color: isActive ? KColors.primary : Colors.white,

borderRadius: BorderRadius.circular(7),

),

padding: EdgeInsets.all(16),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Container(

height: 40,

width: 40,

padding: EdgeInsets.all(10),

decoration: BoxDecoration(

color: isActive ? Colors.white : KColors.lightGrey,

borderRadius: BorderRadius.circular(7),

),

child: Image.asset(img),

),

SizedBox(height: 16),

Text(

company,

style: TextStyle(

fontSize: 12,

color: isActive ? Colors.white38 : KColors.subtitle,

),

),

SizedBox(height: 6),

Text(

title,

style: TextStyle(

fontSize: 14,

color: isActive ? Colors.white : KColors.title,

fontWeight: FontWeight.bold,

),

),

SizedBox(height: 6),

Text(

sub,

style: TextStyle(

fontSize: 12,

color: isActive ? Colors.white38 : KColors.subtitle,

),

),

],

),

),

),

),

);

}

Widget \_recentPostedJob(BuildContext context) {

return Container(

margin: EdgeInsets.symmetric(vertical: 12, horizontal: 16),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(

"Recent posted",

style: TextStyle(fontWeight: FontWeight.bold, color: KColors.title),

),

\_jobCard(context,

img: Images.gitlab,

title: "Gitlab",

subtitle: "UX Designer",

salery: "\$78,000"),

\_jobCard(context,

img: Images.bitbucket,

title: "Bitbucket",

subtitle: "UX Designer",

salery: "\$45,000"),

\_jobCard(context,

img: Images.slack,

title: "Slack",

subtitle: "UX Designer",

salery: "\$65,000"),

\_jobCard(context,

img: Images.dropbox,

title: "Dropbox",

subtitle: "UX Designer",

salery: "\$95,000"),

],

),

);

}

Widget \_jobCard(

BuildContext context, {

String img,

String title,

String subtitle,

String salery,

}) {

return GestureDetector(

onTap: () {

Navigator.push(context, JobDetailPage.getJobDetail());

},

child: Container(

padding: EdgeInsets.symmetric(horizontal: 16, vertical: 12),

margin: EdgeInsets.symmetric(vertical: 6),

decoration: BoxDecoration(color: Colors.white),

child: Row(

children: [

Container(

height: 40,

width: 40,

padding: EdgeInsets.all(8),

decoration: BoxDecoration(

color: KColors.lightGrey,

borderRadius: BorderRadius.circular(4),

),

child: Image.asset(img),

),

SizedBox(width: 10),

Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(

title,

style: TextStyle(fontSize: 12, color: KColors.subtitle),

),

Text(

subtitle,

style: TextStyle(

fontSize: 14,

color: KColors.title,

fontWeight: FontWeight.bold),

)

],

)

],

),

),

);

}

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: KColors.background,

bottomNavigationBar: BottomMenuBar(),

body: SafeArea(

child: Container(

width: MediaQuery.of(context).size.width,

child: SingleChildScrollView(

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

\_appBar(context),

\_header(context),

\_recommendedSection(context),

\_recentPostedJob(context)

],

),

),

),

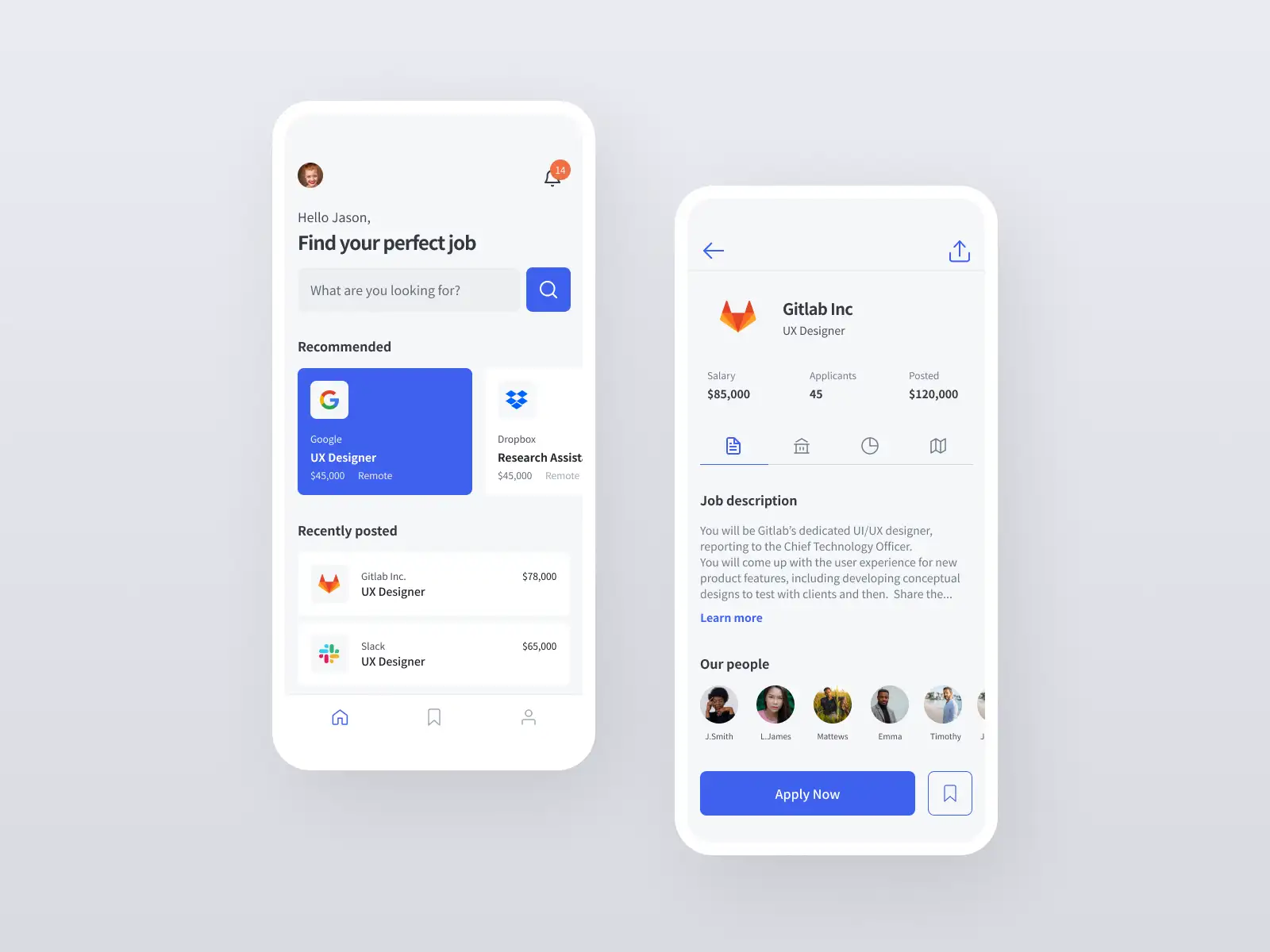
),

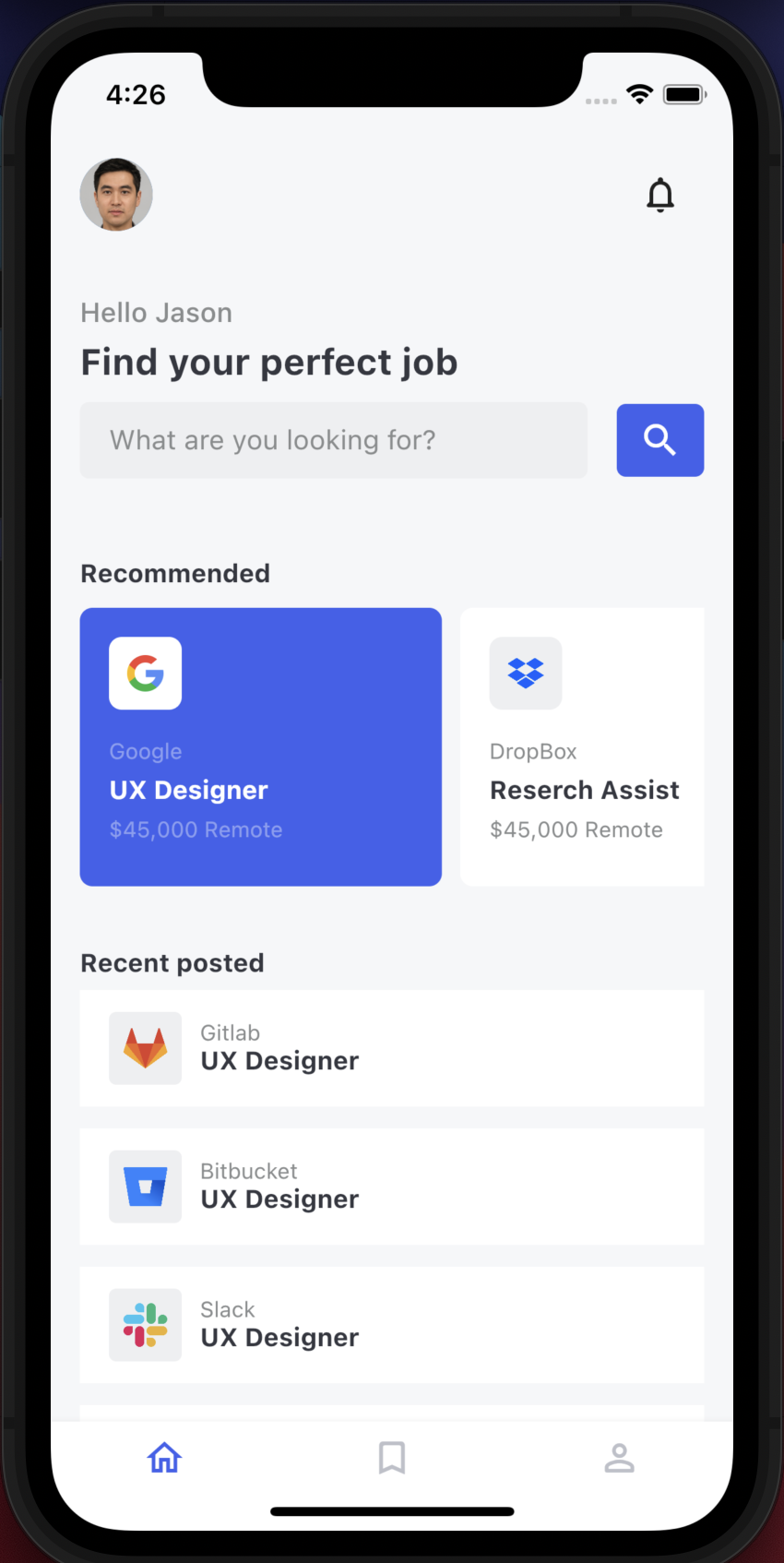
);

}

}

**D. SAMPLE INPUT:**

****

****

