Chapter 2
Survey of Technologies

I. IDEs

1. Android Studio

It is the official integrated development environment (IDE) for developing android apps built on JetBrains's IntelliJ IDE. It uses a Gradle-based build system, Android Emulator, code templates and GitHub integration. It is a one-stop place for development of android apps. It supports various programming languages such as Java, Kotlin or C++, while also supporting various frameworks. It receives the latest updates directly from Google keeping it up-to-date with the trends. It also has support for a large array of third-party plugins. Superior android development, project structure, code completion and refactoring, emulation, etc. are some of the many advantages and features of Android Studio.

2. Eclipse

It is an IDE used primarily for Java development. However, it supports various other languages like C, C++, JavaScript to name a few. For several years, a version of Eclipse with an Android plug-in was recommended, But Google ceased support for this plug-in, causing developers to shift to Android Studio for Android development.

3. NetBeans

It was originally developed by a student of Prague University. It is similar to Eclipse and is primarily known as Java IDE. Android development is supported using plug-ins which are not supported to a large extent today.

4. Visual Studio Code

Very commonly referred to as VS Code, it is an open-source code editor made by Microsoft with the Electron Framework for Windows, Linux and MacOS. It includes vast number of features such as support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring and embedded Git. Users can change the theme, customize keyboard shortcuts and preferences and install extensions that add functionality. A notable feature is the ability to create extensions that add support for new languages, themes, debuggers, etc. via plugins such as the Dart plugin which VS Code utilizes when used to develop Dart apps for which the Flutter framework is used. Good and extensive plugin library is also a very prominent feature of VS Code.

Why have I selected Visual Studio Code?

Currently, I chose Visual Studio Code due to its lightweight nature and the vast number of extensions that come along with it that make the development process much easier. Visual Studio Code takes much less time to set up and open the project as compared to Android Studio which consumes a lot of the computer's resources. While being lightweight, it provides all the necessary features such as debugging and if any features are required, they are available as an extension.

II. Frameworks

1. React Native

It is an open-source cross-platform development framework developed and maintained by Facebook. It uses React, a flexible JavaScript Library to build apps for both Android and iOS. It makes app development much easier and faster by having a single codebase and uses native platform specific built-in components and APIs that give great performance. Having a single codebase reduces development time and cost while having elegant interface across platforms.

2. Flutter

Developed by Google, it is also a free, open-source mobile framework for developing applications. It simplifies the multi-platform development process to craft excellent native interfaces. Flutter is written in Dart language and is a method to implement hybrid app development. Using a single codebase, it uses Google's rendering engine called Skia to develop visuals. It boasts a hot reload functionality which allows for continuous testing without having to restart applications. It is used to develop fast, high- quality applications for iOS and Android in record time from a single codebase.

3. Xamarin

It is also an open-source platform for building modern applications for iOS, Android and Windows applications with .NET. It features a friendly development environment with an abstraction layer that manages communication of shared code with underlying platform code. It allows developers to write all the business logic in a single language while achieving native performance, look and feel on each platform. Its major unique feature is that it uses .NET and C# for cross-platform applications.

4. Ionic

It is designed for developers who are familiar with web development and want to develop hybrid and interactive mobile apps. Its rich and complete set of elements, gestures, animations and software tools enable the developers to develop high quality mobile, desktop and Progressive Web Apps from a single codebase. It is easily integrable with Angular JS and utilizes functionalities like Bluetooth, fingerprint authentication, etc. and also uses emulators, live reload and logging to offer amazing performance. It also uses Cordova plugins to access camera, GPS, etc. in an easy manner. It is the most developer-friendly framework and uses HTML, CSS and JavaScript.

5. Apache Cordova

It makes use of HTML, CSS and JavaScript for building mobile applications. The open-source platform consists of a set of pre-defined plugins that provide access to the device's camera, GPS, file system, etc. The developers can develop apps that are compatible for more than one platform without re-implementing it with the language of each platform. The resulting applications are hybrid, i.e. they are neither truly native nor purely web-based. This causes the applications to run slower than native applications. They are not Web apps because they are not packaged as apps for distribution. The slow or reduced performance is a major drawback however it allows developers to add more functionality using JavaScript.

Why I will be using Flutter?

Flutter is a cross-platform framework developed and maintained by Google themselves. This ensures that the latest features and updates are made available at the earliest, helping the apps to stay up-to-date with the current security, technical and market requirements.

Also, a separate website for iOS users was necessary due to unfeasible cost of publishing apps on the App Store and hence Flutter was chosen that could allow this to be possible using a single codebase without the need for writing code in a separate language. What made Flutter stand out is the large library of components which are highly customizable and are very frequently seen in many of Google's own applications today, hence providing the users with a very familiar user interface. My past experience in developing an Android app also influenced this decision as I have worked with some of these components along with the knowledge of the shortcomings of the previous project, the development approach to be taken and how to implement it which is clearly documented by the official Android team themselves both of which are a part of Google**.**

III. Databases

1. MySQL

It is the most popular open-source relational database management system. Its relational nature helps to organize the data into one or more data tables. The structured data can be inserted, extracted and modified. Its major features are its ability to manage users, allow for network access, facilitating testing and creation of backups. Query caching, Unicode support, multiple storage engines, SQL support are some of its prominent features.

2. PostgreSQL

It is an advanced, enterprise class open-source relational database that supports both SQL (relational) and JSON (non-relational) querying. It is highly stable, backed by more than 20 years of community development which has contributed to its high levels of resilience, integrity and correction. It also has a rich support for advanced data types and performance optimization. However, it is more power-hungry and performs slower when compared to MySQL.

3. SQLite

It is an embedded, file-based RDMS. The application does not run under a separate server process. The serverless architecture enables the database to be cross-platform compatible. The SQLite adheres to ACID properties to safeguard transactions against memory allocation failures. Its compact libraries and small footprint make it fit for applications that do not require a heavy database and because it is stores locally, it can cache data quickly and easily without delaying.

4. Firebase

It is a backend-as-a-service. It is backed by Google and is an application development software that enables developers to develop iOS, Android and Web Apps. The cloud computing services include hosting databases, services, authentication, and integration of applications, including Android, iOS, JavaScript, PHP, C++, etc. It uses NoSQL (not only SQL) as a real-time database. Its blazing speed, cloud-based nature and its ability to handle large sets of data along with numerous features make it a viable option.

5. AWS Dynamo DB

As part of a larger ecosystem of Amazon Web Services (AWS), this database is a NoSQL database that is known for its speed and efficiency when it comes to retrieval of information and data from the system. It is a key-value database, so there are variety of data types that can be stored within this system. It is a highly scalable and a complex database system for developers that are working with the applications that need to manage big user data and constant engagement.

6. MongoDB

It is the most popular NoSQL database and offers many features geared towards the development of mobile applications. As a document-based database, it is proficient with the JSON data-interchange format, making the storage of web pages and other documents like chat logs and messages. It is also highly scalable and has the capacity to grow with the changing needs of the application.

Why I am using MySQL for my application?

I will be using MySQL database because my college infrastructure primarily uses MySQL. Hence, the software and technologies required for the database are already present. This also makes it easier for the people looking after the database and is something that they are familiar with. Also, MySQL is known for its widespread popularity, its speed and its relational nature. It can be accessed via a server and can be stored or hosted remotely easily which ensures the availability of the app at any given time. The data is stored in a relational manner, ie it is easier to understand as it is stored in a structured format.

IV. Other

1. PHP

PHP stands for Hypertext Pre-processor and is a widely used open-source general purpose scripting language. The server-side scripting language is used to perform operations and access databases with the most popular one being MySQL. Using PHP, the data can be inserted and also fetched from the databases making the applications dynamic.

2. Room Library

It is a persistent library that allows fluent access to the database by creating an abstract layer over SQLite. It simplifies the process of adding a structured SQL database to the application.

3. Firebase

This backend-as-a-service includes many features like authentication of email, phone numbers by sending OTP to the users. Also, firebase offers services to send messages to other users as well as send push notifications when required. It offers a vast library of devices on which the application can be simulated and hence perform testing of the applications.

4. Flask

Flask is a lightweight and versatile Python web framework that is used to develop web application quickly and easily using Flask libraries. It is known for its simplicity, flexibility and adaptability. Flask follows a microservices architecture and is it does not have any database abstraction layer, form validation or any other pre-existing components. With its rich support for Python libraries, Flask is a popular choice for implementing server-side Artificial Intelligence features, building APIs, microservices, and web applications.

5. Express.js

Express.js, often referred to as Express, is a lightweight and flexible web application framework for Node.js. It simplifies the process of building web applications and RESTful APIs with a minimalist approach, offering routing, middleware support, HTTP utility methods, and versatility for various application needs. Express has a growing community, scalable architecture, extensive documentation, provides high performance and is the go-to choice for server framework when using Node.js.

6. Node.js

Node.js is an open-source, cross-platform runtime environment that enables developers to build server-side and network applications using JavaScript. Node.js, often referred to simply as Node, is known for its event-driven, non-blocking I/O model, hence providing exceptional speed and efficiency. It is very popularly used to develop highly scalable and real-time applications such as web servers, APIs, microservices, and more. It has a wide range of packages and libraries, and its single-threaded, event-driven architecture makes it suitable for handling a large number of requests, which makes it an excellent choice for applications requiring speed, responsiveness, and scalability.

7. Google Drive

Google Drive is a cloud-based file storage and synchronization service provided by Google. It offers a secure and convenient way to store, access, and share various types of digital content, including documents, images, videos, and more. The service provides a substantial amount of free storage space, with options to upgrade to larger storage plans if needed. Google Drive simplifies file management and sharing, offering a powerful and user-friendly solution for personal and professional use. Using its API, the developers can leverage its benefits and integrate it with their application to store images, videos, etc.

Why Firebase?

I would Firebase Services to send push notifications, authenticating users via email and phone number, etc. Firebase is developed by Google and hence integrates smoothly with Android Studio IDE.

Why Flask?

I would be mainly using Flask to implement Artificial Intelligence features such as looking for matching items. The rich support of libraries in Python for Artificial Intelligence related tasks, its lightweight and flexible nature make Flask the best choice to implement, maintain and scale the APIs easily.

Why Express.js?

As I would be implementing a chatting feature which requires quick response to user interactions, Express.js ensures this speed and efficiency. Socket.IO, a technology that would be used for the chatting feature, works seamlessly with Express.js as both are based on JavaScript. Also, it supports middleware and has an abundance of resources to create APIs. Hence to implement Socket.IO, I would be using Express.js.

Why Node.js?

Express.js is a web application framework that is designed to work with Node.js. Without Node.js, implementing Express.js is not possible as Node.js is the JavaScript runtime environment that executes JavaScript code on the server-side and Express.js heavily relies on Node.js for its core functionality. Hence, I would be using Node.js.

Why Google Drive?

I would primarily use Google Drive as an image storage solution. It is secure, safe and reliable as it provides features to protect stored data such as encryption and access controls. And with the help of Google Drive API, it enables to programmatically interact with stored files and to store files. In the free tier as well, it provides ample free storage of 15GB and this proves to be cost-effective.