

# **Laundry Management System Application**

Submitted in partial fulfillment of the requirements for the award  
of Bachelor of Engineering Degree in  
Computer Science and Engineering

By

**SIDDHA MOHANENDRA (37110716)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND  
TECHNOLOGY**

**(DEEMED TO BE UNIVERSITY)**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

**CHENNAI – 600119, TAMILNADU.**

**MARCH - 2021**



# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**



**Accredited with Grade “A” by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119**

**[www.sathyabamauniversity.ac.in](http://www.sathyabamauniversity.ac.in)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

### **BONAFIDE CERTIFICATE**

This is to certify that this project report is the bonafide work of **SIDDHA MOHANENDRA (17SCS0436)**. Who carried out the project entitled “**LAUNDARY MANAGEMENT SYSTEM APPLICATION**” under my supervision from **August 2020 to March 2021**.

**Internal Guide**

**Dr.T.PREM JACOB.,BE.,ME.,Ph.D**

**Head of the Department**

**Submitted for Viva voce Examination held on\_\_\_\_\_**

**Internal Examiner**

**External Examiner**

## **DECLARATION**

I **SIDDHA MOHANENDRA** here by declare that the Project Report entitled "**LAUNDARY MANAGEMENT SYSTEM APPLICATION**" is done by me under the guidance of **Dr.T.PREM JACOB.,BE.,ME., Ph.D** Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

**DATE:**

**PLACE: CHENNAI**

**SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am are grateful to them.

I convey my thanks to **Dr.T.SASIKALA, M.E., Ph.D., Dean, School of Computing** and **Dr. S. VIGNESWARI, M.E., Ph.D.,Head of the Department, Department of Computer Science and Engineering** for providing me the necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guid **Dr.T.PREM JACOB.,BE.,ME., Ph.D** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the department of COMPUTER SCIENCE AND ENGINEERING who were helpful in many ways for the completion of project.

## **ABSTRACT**

We present the design and implementation of a laundry management system (LMS) used in a laundry establishment. Laundry firms are usually faced with difficulties in keeping detailed records of customers clothing; this little problem as seen to most laundry firms is highly discouraging as customers are filled with disappointments, arising from issues such as customer clothes mix-ups and untimely retrieval of clothes. The aim of this application is to determine the number of clothes collected, in relation to their owners, as this also helps the users fix a date for the collection of their clothes. Also customer's information is secured, as a specific id is allocated per registration to avoid contrasting information.

## TABLE OF CONTENTS

| CHAPTER NO | TITLE                                           | PAGE NO     |
|------------|-------------------------------------------------|-------------|
|            | <b>ABSTRACT</b>                                 | <b>v</b>    |
|            | <b>LIST OF FIGURES</b>                          | <b>viii</b> |
|            | <b>LIST OF ABEVERSACTIONS</b>                   | <b>ix</b>   |
| <b>1</b>   | <b>INTRODUCTION</b>                             | <b>1</b>    |
|            | 1.1 INSIGHT TO PROJECT                          | 1           |
|            | 1.2 STATEMENT OF THE PROBLEM                    | 2           |
|            | 1.3 LAUNDRY APP SERVICE WORK                    | 2           |
|            | 1.4 IMPORTANCE OF LAUNDRY MANAGEMENT<br>SYSYTEM | 2           |
| <b>2</b>   | <b>LITERATURE SURVEY</b>                        | <b>3</b>    |
|            | 2.1 INTRODUCTION TO LAUNDRY                     | 3           |
|            | 2.2 TYPES OF LAUNDRY                            | 4           |
|            | 2.3 INTODUCTION TO ONLINE MARKETING             | 5           |
|            | 2.4 BENEFITS OF E-COMMERCE TO CUSTOMERS         | 7           |
|            | 2.4.1 BENEFITS OF E-COMMERCE TO SOCIETY         | 7           |
|            | 2.4.2 LIMITATIONS OF E-COMMERCE                 | 8           |
|            | 2.4.3 LIMITATIONS OF E-COMMERCE TO<br>COUSTMERS | 8           |
|            | 2.4.4 LIMITATIONS OF E-COMMERCE TO<br>SOCIETY   | 9           |
|            | 2.5 INTRODUCTION TO MANAGEMENT                  | 9           |
| <b>3</b>   | <b>AIM AND SCOPE</b>                            | <b>11</b>   |
|            | 3.1 AIM                                         | 11          |
|            | 3.1.1 EXISTING SYSYTEM                          | 11          |
|            | 3.1.2 PROPOSED SYSTEM                           | 11          |
|            | 3.2 SCOPE                                       | 12          |
|            | 3.2.1 USER                                      | 12          |
|            | 3.2.2 MODULE                                    | 13          |
| <b>4</b>   | <b>SYSTEM ANALYSIS AND METHODOLOGY</b>          | <b>14</b>   |
|            | 4.1 SYSTEM ANALYSIS                             | 14          |

|          |                                  |           |
|----------|----------------------------------|-----------|
|          | 4.2 USER REQUIREMENTS            | 14        |
|          | 4.3 SOFTWARE REQUIREMENTS        | 15        |
| <b>5</b> | <b>RESULT AND DISCUSSION</b>     | <b>26</b> |
|          | 5.1 ANDROID STUDIO AND EMULATOR  | 26        |
|          | 5.2 USER MODULE AND USER SIGN UP | 28        |
|          | 5.3 ADMIN MODULE DASHBOARD       | 28        |
| <b>6</b> | <b>CONCLUSION AND SUMMARY</b>    | <b>33</b> |
|          | 6.1 CONCUSION                    | 33        |
|          | 6.2 SUMMARY                      | 33        |
|          | <b>REFERENCES</b>                | <b>35</b> |
|          | <b>APPENDIX</b>                  | <b>36</b> |
|          | A. SOURCE CODE                   | 36        |

## LIST OF FIGURES

| FIGURE NO | FIGURE NAME                           | PAGE NO |
|-----------|---------------------------------------|---------|
| 3.1       | DIAGRAM OF THE FRONT END AND BACK END | 12      |
| 4.1       | THE PROJECT FILES IN ANDRIOD VIEW     | 16      |
| 4.2       | SHOWING A LAYOUT FILE WITH A PROBLEM  | 17      |
| 4.3       | THE ANDROID STUDIO MAIN WINDOW        | 17      |
| 4.4       | CODE BEFORE FORMATTING                | 19      |
| 4.5       | CODE AFTER FORMATING                  | 19      |
| 4.6       | AN INLINE VARIABLE VALUE              | 22      |
| 5.1       | ANDRIOD STUDIO ANF EMULATOR           | 26      |
| 5.2       | USER MODULE USER SIGN UP              | 28      |
| 5.3       | ADMIN MODULE DASHBOARD                | 29      |
| 5.4       | ORDERD ITEMS                          | 29      |
| 5.5       | LAUNDRY PRICE                         | 30      |
| 5.6       | ORDER DETAILS                         | 30      |
| 5.7       | PRICES OF ORDERED ITEMS               | 31      |
| 5.8       | REAL TIME DATABASE                    | 31      |

## LIST OF ABBREVIATIONS

| ABBREVIATION | EXPANSION                                |
|--------------|------------------------------------------|
| LMS          | LAUNDRY MANAGEMENT SYSTEM                |
| IDE          | INTEGRATED DEVELOPMENT ENVIRONMENT       |
| VCS's        | VERSION CONTROL SYSTEMS                  |
| APK          | ANDROID PACKAGE KIT                      |
| UG           | UNRESTRICTED GUEST                       |
| WHPX         | WINDOWS HYPERVISOR PLATFORM              |
| SSSE3        | SUPPLEMENTAL STREAMING SIMD EXTENSIONS 3 |
| XD           | EXECUTE DISABLE                          |
| AVD          | ANDROID VIRTUAL DEVICE                   |

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 INSIGHT TO PROJECT**

Laundry firms currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the laundry management infrastructure. Often information (on forms) is incomplete or does not follow management standards. Records are often lost in transit during computation requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the laundry firm data and may lead to inconsistencies in data in various data stores.

A significant part of the operation of any laundry firm involves the acquisition, management and timely retrieval of great volumes of information. This information typically involves; customer personal information and clothing records history, user information, price of delivery and retrieval period, users scheduling as regards customers details and dealings in service rendered, also our products package waiting list. All of this information must be managed in an efficient and cost wise fashion so that the organization resources may be effectively utilized.

The goal of laundry management system is to automate the management of the laundry firm making it more efficient and error free. It aims at standardizing data, consolidating data ensuring data integrity and reducing inconsistencies, through the use of highly computerized process that is stress free, reliable and quick through the use of asp.net computer programming language and SQL database application to both the users and the staff in charge of the registration and laundry management processes. XML would be at the front-end and provide the graphical user interface that relates with the user, while the SQL database will be at the back-end to handle the data storage process.

An essential piece of the activity of any wear firm includes the procure, the board and opportune recovery of extraordinary volumes of knowledge. This knowledge usually includes; shopper individual knowledge and clothes records history, shopper knowledge, price of conveyance and recovery amount, purchasers booking as

respects purchasers subtleties and dealings in facilitate delivered, in addition our things bundle holding up list. All of this knowledge ought to be overseen in an exceedingly productive and value savvy vogue that the association assets may well be viably used.

The objective of Vesture companies is it's board framework is to mechanize the administration of the Vesture firm making it more productive and mistake free. It targets normalizing information, solidifying information guaranteeing information respectability and decreasing irregularities, using profoundly mechanized interaction that is peaceful, dependable and fast using asp.net PC programming language and SQL data set application to both the clients and the staff accountable for the staffing and Vesture the executives measures. XML would be at the front-end and give the graphical UI that relates with the client, while the SQL information base will be at the back-finish to deal with the information stockpiling measure.

## **1.2 STATEMENT OF THE PROBLEM**

The experience of washing clothes is not pleasant at all. It's time consuming, whether you do it at home or take it to the Laundromat.

The use of manual system also create an additional workload for staff to keep and obtain the customer and staff information because this information is kept in a different file.

## **1.3 LAUNDRY APP SERVICE WORK:**

A wash and fold laundry service is probably exactly what you are picturing. The professionals at the Laundromat then wash, dry, and fold the laundry for pick up. All you have to do is drop off your dirty laundry and pick up your fresh, clean laundry at a time that is convenient for you.

## **1.4 IMPORTANCE OF LAUNDRY MANAGEMET SYSTEM:**

Laundry cleaning servies for both clothes and curtains specialize in quality care for laundry. Because they are in the industry, laundry services understand the importance of using these differen cleaning methods. They will not only gets your laundry clean but also prevent them from shrinkage and colour bleeding these are the importance of landuary management system.

## **CHAPTER 2**

### **LITERATURE**

### **SURVEY**

#### **2.1 INTRODUCTION TO LAUNDRY**

Laundry is washing of clothing and lines. Laundry processes are often done in a room reserved for that purpose; in an individual home this is referred as laundry room. An apartment building or student hall of residence may have a shared laundry facility such as tvättstuga. A stand -alone business is referred to as a laundrette (laundromat). The material that is being washed,or has been launderd, is also generally referred to as laundry.

Laundry was first done in watercourse, letting the water carry away the material which could cause stains and smells laundry is still done this way in some less industrialized area and rural regions. Agitation help remove the dirt, so the laundry is often rubbed, twisted, or slapped against flat rocks. Wooden bats or clubs could be used help with beating the dirt out. These were often called washing beetles or bats and could be used by the waterside on a rock (a beetling -stone) ,on a block (battling-block),or on a washboard. They were once common across Europe and were also been identified in Japan. Wooden or stone scrubbing surface set up near a water supply or portable washboards, including factory-made corrugated glass or metal ones, gradually replaced rocks as a surface for losing soil. Once clean,the clothes were wrung out -twisted to remove most of the water. Then they were hung up on poles or clothesline to air dry, or sometimes just spread out on clegrass. Before the advent of the washing machine, laundry was often done in a communal setting. In poor parts of the world today, laundry is still done beside a river or lake. Villages across Europe that could afford it built a wash-house. Water was channeled from a stream or spring and fed into a building, possibly just a roof with no walls. This wash-house usually contained two basins - one for washing and the other for rinsing - through which the water was constantly flowing, as well as a stone lip inclined towards the water against which the washers could beat the clothes. Such facilities were much more comfortable than washing in a watercourse because the launderers could work standing up instead of on their knees, and were protected from inclement weather. Also, they didn't have to go far, as the facilities were usually at hand in the village or at the edge of a town. A poser could be used to agitate clothes in a tub (according to Old and Interesting). These facilities were public and

available to all families, and usually used by the entire village. Many of these village wash-houses are still standing, historic structures with no obvious modern purpose. This job was reserved for women, who washed all their family's laundry. Washerwomen took in the laundry of others, charging by the piece. As such, wash-houses were an obligatory stop in many women's weekly lives and became a sort of institution or meeting place. It was a women-only space where they could discuss issues or simply chat. Indeed, this tradition is reflected in the Catalan idiom "fer safareig" (literally, "to do the laundry"), which means to gossip.

European cities also had public wash-houses. The city authorities wanted to give the poorer population, who would otherwise not have access to laundry facilities, the opportunity to wash their clothes. Sometimes these facilities were combined with baths. The aim was to foster hygiene and thus reduce outbreaks of epidemics.

A laundry-worker took sopping wet clothing and cranked it through the mangle, compressing the cloth and expelling the excess water. The mangle was much quicker than hand twisting. It was a variation on the box mangle used primarily for pressing and smoothing cloth. Meanwhile, 19th century inventors further mechanized the laundry process with various hand-operated washing machines. Most involved turning a handle to move paddles inside a tub. Then some early 20th century machines used an electrically powered agitator to replace tedious hand rubbing against a washboard. Many of these were simply a tub on legs, with a hand-operated mangle on top. Later the mangle too was electrically powered, then replaced by a perforated double tub, which spun out the excess water in a spin cycle. Laundry drying was also mechanized, with clothes dryers

## **2.2 Types of Laundry**

Laundry is categorized into different type due to their size, scale, products offered, Store Format and Trends While people use the terms "Industrial", "Commercial" and "Residential" interchangeably to refer to laundry services, industry watchers offer more specific guidelines about different types of Laundry. "Industrial type" is on the larger end of this spectrum and carry a diverse mix of machines and general merchandise. Nomenclature is not always uniform Financial Institutions Fund places Wal-Mart in the same category as supermarkets and as well running laundry services.

**Industrial Laundry:** This type is for the big guys. Usually utilizes the use of a tunnel washer and/or a heavy duty front load washer with big capacity, about 50kgs up per

machine. It would require big investment. Clients to look for if you have this type of laundry would be institutional like hospital, hotel, and motel Spa and etc.

**Industrial Laundry:** This type is for the big guys. Usually utilizes the use of a tunnel washer and/or a heavy duty front load washer with big capacity, about 50kgs up per machine. It would require big investment. Clients to look for if you have this type of laundry would be institutional like hospital, hotel, and motel. Spa and etc

### **Operating a commercial laundry**

There are two kinds:

1. **Stand-alone** - this means all your machines are within your business premises. In Asia, the lead time would be 1- 3 days to do the laundry. Other countries would be hours only if the units are coin operated
2. **Pick-up station** - if you are still uncertain if you want to go full time and let go of your hard earned money. You might want to be a partner of a stand-alone shop owner. The commission will have to be agreed upon by the two parties. By doing this kind, you are actually building up your own market. If you feel you have already enough market, then that would be the time to go Stand Alone.

**Commercial laundry:** this makes use, of course. They are commonly found machine in the market. In Asia, the way the use it, people though they may be wrong, the built of the units are other than plastic. Mostly is aluminum with metal base. In countries other than Asia, the common brands are whirlpool, Maytag, Samsung etc. The target clients are mostly walk-ins.

**Residential laundry :** As implied, the operation uses an ordinary unit which is usually made of plastic. Not durable for a 24/7 operation. Though if you are in a start-up and would like to test market, then fine and go. However, it's not recommend the use of residential machines in a laundry business. Basically the type of Laundry is determined by the machines to be used and targeted clients.

### **2.3 Introduction to online Marketing (E- commerce)**

The internet marketing has been active for a long time now, the cumulative events occurring in online marketing is leading up to where we are now it have impacted the entire globe faster than any marketing revolution in history.

Over the past decade or so, supermarkets, laundries and other grocery retailers have continued to invest significantly into broadening their Internet presence and expanding the number of channels through which their goods or services are

sold/made. Key Note estimates that sales of groceries transacted via online channels observed double-digit growth between 2007 and 2011, increasing by 127% overall.

One of the major trends to have driven growth within the Internet grocery market is m-commerce that is sales made via mobile channels, i.e. smart phones and tablet computers. The increasing popularity of smart phones and tablets among customers has resulted in a whole host of retailers investing significant sums of money into mobile sales platforms, as well as downloadable applications (apps'), which offer a more interactive and personalized shopping and Services experience.

Despite the growth of online grocers in recent years, online spending still accounts for a relatively small proportion of the overall Internet grocery market, with just 3.9% of total grocery sales estimated to have been transacted via e-commerce and m-commerce channels. However, the share of the total grocery market represented by online grocers has continued to increase year-on-year since at least 2007, when their market share stood at just 2.1%.

Key Note expects the Internet grocery market to continue to go from strength to strength over the forthcoming years and has forecast year-on-year double-digit growth for 2012 to 2016. The rising uptake of Internet-connected mobile devices, such as smart phones and tablets, should boost services and sales transacted via m-commerce channels, while continued Government investment in the rollout of superfast broadband, alongside the introduction of the UK's first 4G mobile network, will also help to boost Internet activity and the use of e-commerce services throughout the country.

Online marketing can broadly be defined as the processes or areas involved in the running and operation of an organization that are electronic or digital in nature. These include direct business activities such as marketing, sales and human resource management but also indirect activities such as business process re-engineering and change management, which impact on the improvement in efficiency and integration of business processes and activities.

In 1994, spending for internet marketing totaled nearly nothing, but increased to over \$300 million in 1995. Now, little more than a decade later, marketing spending and internet marketing business has exploded to nearly \$200 billion (according to Forrester Research). Today, it's hard to believe in having an organization which doesn't have some kind of online presence.

When the internet was first introduced in the early 90s, it wasn't considered to be an advertising medium at all. Instead, the internet was treated as a tool for exchanging emails and digital information, but wasn't yet considered valuable for reaching customers. However, it wasn't long before marketing pioneers began to see the potential for internet marketing business as millions of web surfers logging on each day to find valuable and relevant information. Within just a few years, informative and educational marketing, as well as graphically enticing banner ads began to be show up. It wasn't long before results began to flood in which proved the value of the internet marketplace to even the most skeptical advertisers.

Factors that affect online marketing are as follows:

- Technological Factors
- Social Factors
- Economic Factors.

## **2.4 Benefits of E-Commerce to Customers**

**24/7 access:** It enables customers to shop or conduct other transactions 24 hours a day, all year round from almost any location. For example checking balances, making payments, obtaining travel tickets and other information. In one case a pop star set up web cameras in every room in his house, so that he could check the status of his home by logging onto the Internet when he was away from home on tour.

**More choices:** Customers not only have a whole range of services that they can choose from and customize, but also an international selection of staffs.

**Price comparison:** Customers can select services around the world and conduct comparisons either directly by visiting different sites, or by visiting a single site where prices are aggregated from a number of providers and compared (for example [www.moneyextra.co.uk](http://www.moneyextra.co.uk) for financial products and services).

**Improved delivery processes:** can range from the immediate delivery of digitized or electronic goods such as software or audio-visual files by downloading via the Internet, to the on- line tracking of the progress of packages being delivered by mail or courier

### **2.4.1 Benefits of E-Commerce to society**

It enables more flexible working practices, which enhances the quality of life for a whole host of people in society, enabling them to work from home. Not only is this

more convenient and provides happier and less stressful working environments, it also potentially reduces environmental pollution as fewer people have to travel to work regularly.

Enables people in developing countries and rural areas to enjoy and access products, services, information and other people which otherwise would not be so easily available to them.

#### ***2.4.2 Limitations of E-Commerce***

There was much hype surrounding the Internet and e-commerce over the last few years of the twentieth century. Much of it promoted the Internet and e-commerce as the panacea for all ills, which raises the question, are there any limitations of e-commerce and the Internet? Isaac Newton's 3rd Law of Motion, for every action there is an equal and opposite reaction suggests that for all the benefits there are limitations to e-commerce. These again will be dealt with according to the three major stakeholders' organizations, consumers and society.

This includes the following:

Rapidly evolving and changing technology, so there is always a feeling of trying to catch up and not be left behind. Under pressure to innovate and develop business models to exploit the new opportunities which sometimes leads to strategies detrimental to the organization. The ease with which business models can be copied and emulated over the Internet increases that pressure and curtails longer-term competitive advantage.

Facing increased competition from both national and international competitors often leads to price wars and subsequent unsustainable losses for the organization.

There are problems where older business systems cannot communicate with web based and Internet infrastructures, leading to some organizations running almost two independent systems where data cannot be shared. This often leads to having to invest in new systems or an infrastructure, which bridges the different systems. In both cases this is both financially costly as well as disruptive to the efficient running of organizations.

#### ***2.4.3 Limitations of E-Commerce to Customers***

Computing equipment is needed for individuals to participate in the new 'digital' economy, which means an initial capital cost to customers.

A basic technical knowledge is required of both computing equipment and navigation of the Internet and the World Wide Web.

Cost of access to the Internet, whether dial-up or broadband tariffs.

Cost of computing equipment. Not just the initial cost of buying equipment but making sure that the technology is updated regularly to be compatible with the changing requirement of the Internet, websites and applications.

Lack of security and privacy of personal data. There is no real control of data that is collected over the Web or Internet. Data protection laws are not universal and so websites hosted in different countries may or may not have laws which protect privacy of personal data.

Physical contact and relationships are replaced by electronic processes. Customers are unable to touch and feel goods being sold on-line or gauge voices and reactions of human beings.

#### ***2.4.4 Limitations of E-Commerce to society***

**Breakdown in human** interaction: As people become more used to interacting electronically there could be an erosion of personal and social skills which might eventually be detrimental to the world we live in where people are more comfortable interacting with a screen than face to face. **Social division:** There is a potential danger that there will be an increase in the social divide between technical haves and have-nots – so people who do not have technical skills become unable to secure better-paid jobs and could form an underclass with potentially dangerous implications for social stability.

#### **2.5 Introduction to management**

The Term management is the organization and coordination of the activities of a business in order To achieve defined objectives. Management is often included as a factor of production

Along with machines, materials, and money. According to the management guru Peter Drucker (1909-2005), the basic task of management includes both marketing and innovation.

Practice of modern management originates from the 16th century study of low-efficiency and failures of certain enterprises, conducted by the English statesman Sir Thomas More (1478- 1535). Management consists of the interlocking functions of

creating corporate policy and organizing, planning, controlling, and directing an organization's resources in order to achieve the objectives of that policy.

The size of management can range from one person in a small organization to hundreds or thousands of managers in multinational companies. In large organizations, the board of directors defines the policy which is then carried out by the chief executive officer, or CEO. Some people agree that in order to evaluate a company's current and future worth, the most important factors are the quality and experience of the managers.

Management involves the manipulation of the human capital of an enterprise to contribute to the success of the enterprise. This implies effective communication: an enterprise environment (as opposed to a physical or mechanical mechanism), implies human motivation and implies some sort of successful progress or system outcome. As such, management is not the manipulation of a mechanism (machine or automated program), not the herding of animals, and can occur in both a legal as well as illegal enterprise and environment. Based on this, management must have humans, communication, and a positive enterprise endeavor. Plans, measurements, motivational psychological tools, goals, and economic measures (profit, etc.) may or may not be necessary components for there to be management. At first, one views management functionally, such as measuring quantity, adjusting plans, meeting goals. This applies even in situations where planning does not take place. From this perspective, Henri Fayol (1841–1925) considers management to consist of six functions:

1. Forecasting
2. Planning
3. Organizing
4. Commanding
5. Coordinating
6. Controlling

### **3.1 AIM**

C  
H  
A  
P  
T  
E  
R  
  
3  
  
A  
I  
M  
  
A  
N  
D  
  
S  
C  
O  
P  
E

The objective of this work is to implement a management system that will streamline registration process, reduce administrative tasks and paper work so as to improve the registration cycle process flow.

#### **3.1.1 EXISTING SYSTEM**

Laundry firm currently uses a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the Laundry firm management infrastructure. Often information (on forms) is incomplete, or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. This has led to inconsistencies in various data due to large volume of contrasting customer details leading to mix-up of clothes in the laundry firm which thus leads to delay in collecting the clothes back.

### **3.1.2 PROPOSED SYSTEM**

The Laundry Management System is designed for any Laundry firm to replace their existing manual, paper based system. The new system is in form of an e-registration system to control the following; customer information, products, services, users, carts and receipt. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the delay and resources currently required for such tasks as clothes details are bounded to a particular customer with a given id. Since the existing system makes use of tedious administrative tasks, lots paper work and time, in which full information cannot be gotten from busy customers.

The goal of the laundry management system is to provide a computerized process that is stress free, reliable and quick through the use of asp.net computer programming language and SQL database application to the users and staffs in charge of the registration of customers and laundry management processes. JAVA would be at the front-end and provide the graphical user interface that relates with

the user, while the SQL database will be at the back-end to handle the data storage process.

### ***FIG 1: DIAGRAM OF THE FRONT END AND BACK END***

#### **ComputerizedSystem:**

The proposed system will implement the computerized system which can perform a better managing process for the laundry. The data of the laundry service and the customer will kept in the save manner without the problem of losing the data.

#### **System and UserPrivileges:**

System and user privileges will be implemented in the proposed system to setting up the user level for each system user. This function is to provide the limitation of system accessing.

#### **Increase timeperformance:**

The time management is very important for the laundry management to ensure the service performs in better condition and on time. In addition, by using the computerized system, the business process will be more effective and faster.

#### **Justification of theStudy:**

The new system is design to solve problem affecting the manual system in use. It design to computerized information of the laundry thereby relieving both customer and services frommuch stress as experienced from the manualsystem. This will do the analyzing and storing of information either automatically or interactively, it will make use of computerized system to access the information.

### **3.2 Scope:**

#### **3.2.1 User**

The target users for this system are Laundry staffs, managers and system administrator.

**i. Staff**

Staff of Laundry has privileges to insert, view, update and delete customer records. Beside managing service and launder details, staff also has a privilege to calculate payment for each transaction made.

**ii. Manager**

Owner or manager of Laundry shop or company is a person who has privileges to view staff information and update their information.

**iii. Administrator**

Administrator is a person who has responsibility to maintain the system. Administrator has all privileges to this system and may grant the privileges to another system User.

### **3.2.2 Module**

Some of the modules in Laundry Management System (LMS) are:

**iv. LoginModule**

This module allows user to the system by entering a valid username and password.

**v. Customer Module**

This allows staff to add, update, delete and retrieve customer record.

**vi. ServiceModule**

This module is used to record service details which are available for staff of the Laundry to use it.

**vii. Service ApplicationModule**

This module is used to record laundry services to the customers which refers to staff to handle it. The module is specialize to calculate the total amount of payment for services. Furthermore, this module will keep in LMS database.

**viii. UsersModule**

This module allow manager to add, update, delete and retrieve staff information, such as giving them Username and Password to access the database.

**vi Logout**

This module allows the user to Logout the application. Futher operations cannot be performed after user exits.

## **CHAPTER 4**

### **SYSTEM ANALYSIS AND METHODOLOGY**

#### **4.1 System Analysis**

System Analysis and Design System analysis is a method of problem-solving that deals with the breaking down of a system into components parts in order to study how well the individual parts work and interact to accomplish their purpose. It involves the process of enumerating the existing problems, analyzing the proposed system for costs and benefits, analyzing the system and user requirements and considering possible alternative system. System analysis is important in the design of subsequent systems. System design consists of design activities that produce system specifications which satisfy the functional requirements that have been developed in the system analysis process. System design is basically the structural implementation of system analysis. The proposed system is being designed in such a way that users only need to input their customer data which is then entered into a computer database. Customers will be assigned a specific id on registration.

#### **4.2 User Requirements**

To gain access to the laundry management system resources, the user would need:

- 1) A personal computer
  - 2) A username
  - 3) A genuine password
- User-Interface Requirements User interfaces are the registration pages developed for the customers and users to register and manage the items brought. They consist of the following:
- 4) Login page (Username and password)
  - 5) Home page
  - 6) View price
  - 7) View records
  - 8) Register a new user
  - 9) Print Receipt

**4.3 Software Requirements:**Someone who want to build android applications must have android studio and one who want to test application virtually must need emulator preinstalled.

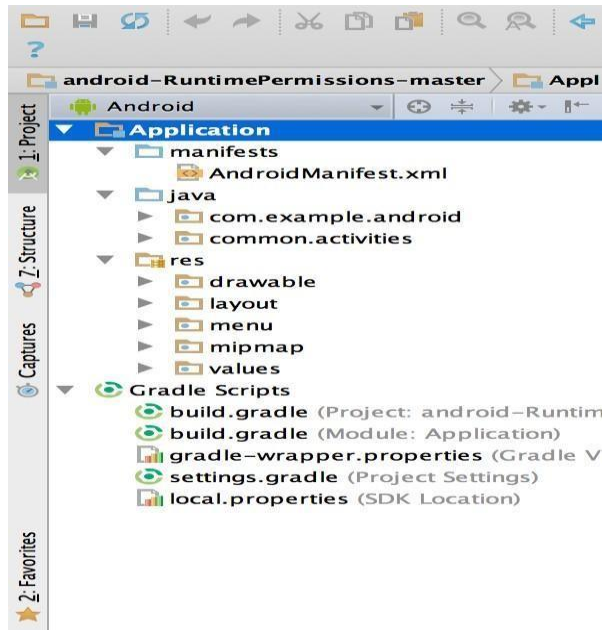
**Android studio:**

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

This page provides an introduction to basic Android Studio features. For a summary of the latest changes, see [Android Studio release notes](#).

## Project structure



**Fig4.1: The project files in Android view.**

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

- Android app modules
- Library modules
- Google App Engine modules

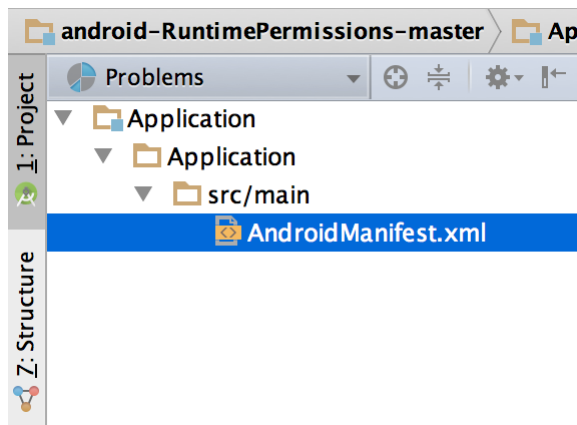
By default, Android Studio displays your project files in the Android project view, as shown in figure 1. This view is organized by modules to provide quick access to your project's key source files.

All the build files are visible at the top level under **Gradle Scripts** and each app module contains the following folders:

- **manifests**: Contains the AndroidManifest.xml file.
- **java**: Contains the Java source code files, including JUnit test code.
- **res**: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select **Project** from the **Project** dropdown (in figure 1, it's showing as **Android**).

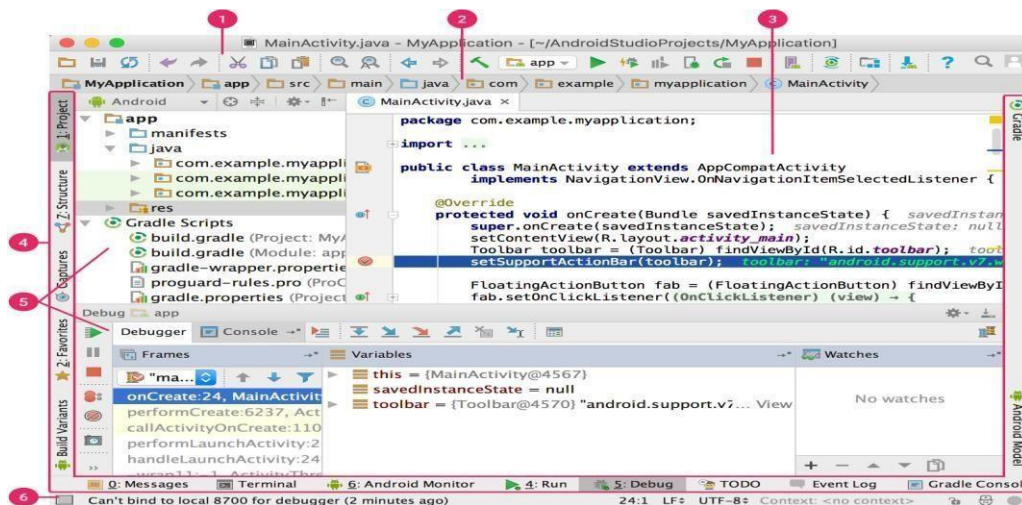
You can also customize the view of the project files to focus on specific aspects of your app development. For example, selecting the **Problems** view of your project displays links to the source files containing any recognized coding and syntax errors, such as a missing XML element closing tag in a layout file.



**Fig4.2: The project files in Problems view, showing a layout file with a problem.**

The user interface

The Android Studio main window is made up of several logical areas identified in figure 3.



**Fig 4.3: The Android Studio main window.**

1. The **toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.
2. The **navigation bar** helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the **Project** window.
3. The **editor window** is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.
4. The **tool window bar** runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.

5. The **tool windows** give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.
6. The **status bar** displays the status of your project and the IDE itself, as well as any warnings or messages.

You can organize the main window to give yourself more screen space by hiding or moving toolbars and tool windows. You can also use keyboard shortcuts to access most IDE features.

At any time, you can search across your source code, databases, actions, elements of the user interface, and so on, by double-pressing the Shift key, or clicking the magnifying glass in the upper right-hand corner of the Android Studio window. This can be very useful if, for example, you are trying to locate a particular IDE action that you have forgotten how to trigger.

### Tool windows

Instead of using preset perspectives, Android Studio follows your context and automatically brings up relevant tool windows as you work. By default, the most commonly used tool windows are pinned to the tool window bar at the edges of the application window.

- To expand or collapse a tool window, click the tool's name in the tool window bar. You can also drag, pin, unpin, attach, and detach tool windows.
- To return to the current default tool window layout, click **Window > Restore Default Layout** or customize your default layout by clicking **Window > Store Current Layout as Default**.
- To show or hide the entire tool window bar, click the window icon  in the bottom left-hand corner of the Android Studio window.
- To locate a specific tool window, hover over the window icon and select the tool window from the menu.

You can also use keyboard shortcuts to open tool windows.

### Navigation

Here are some tips to help you move around Android Studio.

- Switch between your recently accessed files using the *Recent Files* action. Press **Control+E** (**Command+E** on a Mac) to bring up the Recent Files action. By default, the last accessed file is selected. You can also access any tool window through the left column in this action.
- View the structure of the current file using the *File Structure* action. Bring up the File Structure action by pressing **Control+F12** (**Command+F12** on a Mac). Using this action, you can quickly navigate to any part of your current file.
- Search for and navigate to a specific class in your project using the *Navigate to Class* action. Bring up the action by pressing **Control+N** (**Command+O** on a Mac).

Navigate to Class supports sophisticated expressions, including camel humps, paths, line navigate to, middle name matching, and many more. If you call it twice in a row, it shows you the results out of the project classes.

- Navigate to a file or folder using the *Navigate to File* action. Bring up the Navigate to File action by pressing **Control+Shift+N** (**Command+Shift+O** on a Mac). To search for folders rather than files, add a / at the end of your expression.
- Find all the pieces of code referencing the class, method, field, parameter, or statement at the current cursor position by pressing **Alt+F7** (**Option+F7** on a Mac).

### Style and formatting

As you edit, Android Studio automatically applies formatting and styles as specified in your code style settings. You can customize the code style settings by programming language, including specifying conventions for tabs and indents, spaces, wrapping and braces, and blank lines. To customize your code style settings, click **File > Settings > Editor > Code Style (Android Studio > Preferences > Editor > Code Style** on a Mac.) Although the IDE automatically applies formatting as you work, you can also explicitly all the *ReformatCode* action by pressing **Control+Alt+L**(**Opt+Command+L** on a Mac), or auto-indent all lines by pressing **Control+Alt+I** (**Control+Option+I** on a Mac).

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    mActionBar = getSupportActionBar();  
    mActionBar.setDisplayHomeAsUpEnabled(true);  
}
```

**Fig4.4:Code before formatting.**

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    mActionBar = getSupportActionBar();  
    mActionBar.setDisplayHomeAsUpEnabled(true);  
    // Get reference to the drawer layout and set event listener
```

Formatted 7 lines  
Show reformat dialog: ⌘⇧⌘L

**Fig 4.5: Code after formatting.**

## Version control basics

Android Studio supports a variety of version control systems (VCS's), including Git, GitHub, CVS, Mercurial, Subversion, and Google Cloud Source Repositories.

After importing your app into Android Studio, use the Android Studio VCS menu options to enable VCS support for the desired version control system, create a repository, import the new files into version control, and perform other version control operations:

1. From the Android Studio **VCS** menu, click **Enable Version Control Integration**.
2. From the drop-down menu, select a version control system to associate with the project root, and then click **OK**.

The VCS menu now displays a number of version control options based on the system you selected.

## Gradle build system

Android Studio uses Gradle as the foundation of the build system, with more Android-specific capabilities provided by the Android plugin for Gradle. This build system runs as an integrated tool from the Android Studio menu, and independently from the command line. You can use the features of the build system to do the following:

- Customize, configure, and extend the build process.
- Create multiple APKs for your app, with different features using the same project and modules.
- Reuse code and resources across sourcesets.

By employing the flexibility of Gradle, you can achieve all of this without modifying your app's core source files. Android Studio build files are named `build.gradle`. They are plain text files that use Groovy syntax to configure the build with elements

provided by the Android plugin for Gradle. Each project has one top-level build file for the entire project and separate module-level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files. To learn more about the build system and how to configure, see [Configure your build](#).

### **Build variants**

The build system can help you create different versions of the same application from a single project. This is useful when you have both a free version and a paid version of your app, or if you want to distribute multiple APKs for different device configurations on Google Play.

### **Multiple APK support**

Multiple APK support allows you to efficiently create multiple APKs based on screen density or ABI. For example, you can create separate APKs of an app for the hdpi and mdpi screen densities, while still considering them a single variant and allowing them to share test APK, javac, dx, and ProGuard settings.

### **Resource shrinking**

Resource shrinking in Android Studio automatically removes unused resources from your packaged app and library dependencies. For example, if your application is using Google Play services to access Google Drive functionality, and you are not currently using Google Sign-In, then resource shrinking can remove the various drawable assets for the `SignInButton` buttons.

### **Debug and profile tools**

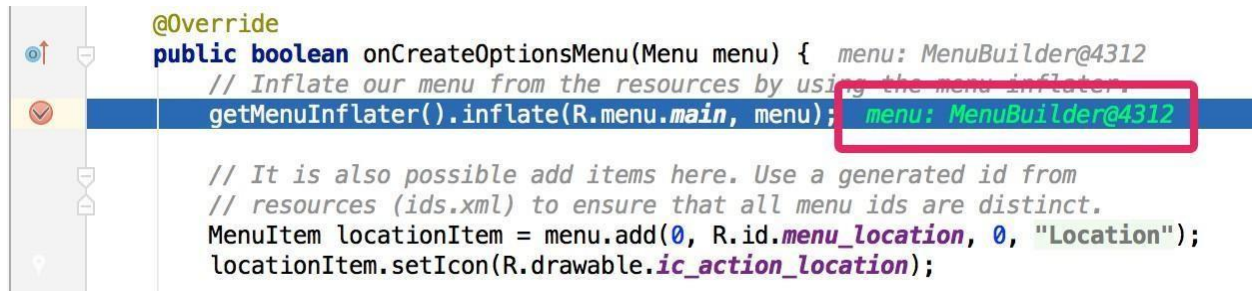
Android Studio assists you in debugging and improving the performance of your code, including inline debugging and performance analysis tools.

### **Inline debugging**

Use inline debugging to enhance your code walk-throughs in the debugger view with inline verification of references, expressions, and variable values. Inline debug information includes:

- Inline variable values

- Referring objects that reference a selected object
- Method return values
- Lambda and operator expressions
- Tooltip values



**Fig 4.6: An inline variable value.**

To enable inline debugging, in the **Debug** window, click **Settings** and select the checkbox for **Show Values Inline**.

### Performance profilers

Android Studio provides performance profilers so you can more easily track your app's memory and CPU usage, find deallocated objects, locate memory leaks, optimize graphics performance, and analyze network requests. With your app running on a device or emulator, open the **Android Profiler** tab.

### Emulator

Requirements and recommendations:

The Android Emulator has additional requirements beyond the basic system requirements for Android Studio, which are described below:

- SDK Tools 26.1.1 or higher
- 64-bit processor
- Windows: CPU with UG (unrestricted guest) support
- HAXM 6.2.1 or later (HAXM 7.2.0 or later recommended)

The use of hardware acceleration has additional requirements on Windows and Linux:

- Intel processor on Windows or Linux: Intel processor with support for Intel VT-x, Intel EM64T (Intel 64), and Execute Disable (XD) Bit functionality
- AMD processor on Linux: AMD processor with support for AMD Virtualization (AMD-V) and Supplemental Streaming SIMD Extensions 3 (SSSE3)
- AMD processor on Windows: Android Studio 3.2 or higher and Windows 10 April 2018 release or higher for Windows Hypervisor Platform (WHPX) functionality

To work with Android 8.1 (API level 27) and higher system images, an attached webcam must have the capability to capture 720p frames.

### **Deprecation for 32-bit Windows systems**

The Android Emulator was deprecated in June 2019 for 32-bit Windows systems. Support for the 32-bit Windows emulator continues until June 2020, including critical bug fixes, but no new features will be added. If you are using the emulator on a 32-bit Windows system, you should plan to migrate to a 64-bit Windows system.

If you are using the emulator on a 32-bit Windows system, you can use the SDK Manager to install the latest version of the emulator for 32-bit Windows.

Install the emulator

To install the Android Emulator, select the **Android Emulator** component in the **SDK Tools** tab of the **SDK Manager**. For instructions, see [Update your tools using the SDK Manager](#).

Android virtual devices

Each instance of the Android Emulator uses an *Android virtual device (AVD)* to specify the Android version and hardware characteristics of the simulated device. To effectively test your app, you should create an AVD that models each device on which your app is designed to run. To create and manage AVDs, use the AVD Manager.

Each AVD functions as an independent device, with its own private storage for user data, SD card, and so on. By default, the emulator stores the user data, SD card data, and cache in a directory specific to that AVD. When you launch the emulator, it loads the user data and SD card data from the AVD directory.

### How to Run an app on the Android Emulator

You can run an app from an Android Studio project, or you can run an app that's been installed on the Android Emulator as you would run any app on a device.

To start the Android Emulator and run an app in your project:

1. In Android Studio, create an Android Virtual Device (AVD) that the emulator can use to install and run your app.
2. In the toolbar, select the AVD that you want to run your app on from the target device drop-down menu.
3. Click **Run**
4. If you receive an error or warning message at the top of the dialog, click the link to correct the problem or to get more information.
5. Some errors you must fix before you can continue, such as certain Hardware Accelerated Execution Manager (Intel HAXM) errors.
6. For macOS, if you see a Warning: No DNS servers found error when starting the emulator, check to see whether you have an `/etc/resolv.conf` file. If you don't have this file, enter the following command in a terminal window:

How to Launch the Android Emulator without first running an app:

To start the emulator:

1. Open the AVD Manager.
2. Double-click an AVD, or click **Run**

The Android Emulator loads.

While the emulator is running, you can run Android Studio projects and choose the emulator as the target device. You can also drag one or more APKs onto the emulator to install them, and then run them.

How to Run the Android Emulator directly in Android Studio: Run the Android Emulator directly in Android Studio to conserve screen real estate, to navigate quickly between the emulator and the editor window using hotkeys, and to organize your IDE and emulator workflow in a single application window.

To run the emulator in Android Studio, make sure you're using Android Studio 4.1 or higher with version 30.0.10 or higher of the Android Emulator, then follow these steps:

1. Click **File > Settings > Tools > Emulator** (or **Android Studio > Preferences > Tools > Emulator on macOS**), then select **Launch in a tool window** and click **OK**.
2. If the Emulator window didn't automatically appear, open it by clicking **View > Tool Windows > Emulator**.
3. Start your virtual device using the **AVD Manager** or by targeting it when running your app.

#### **Limitations:**

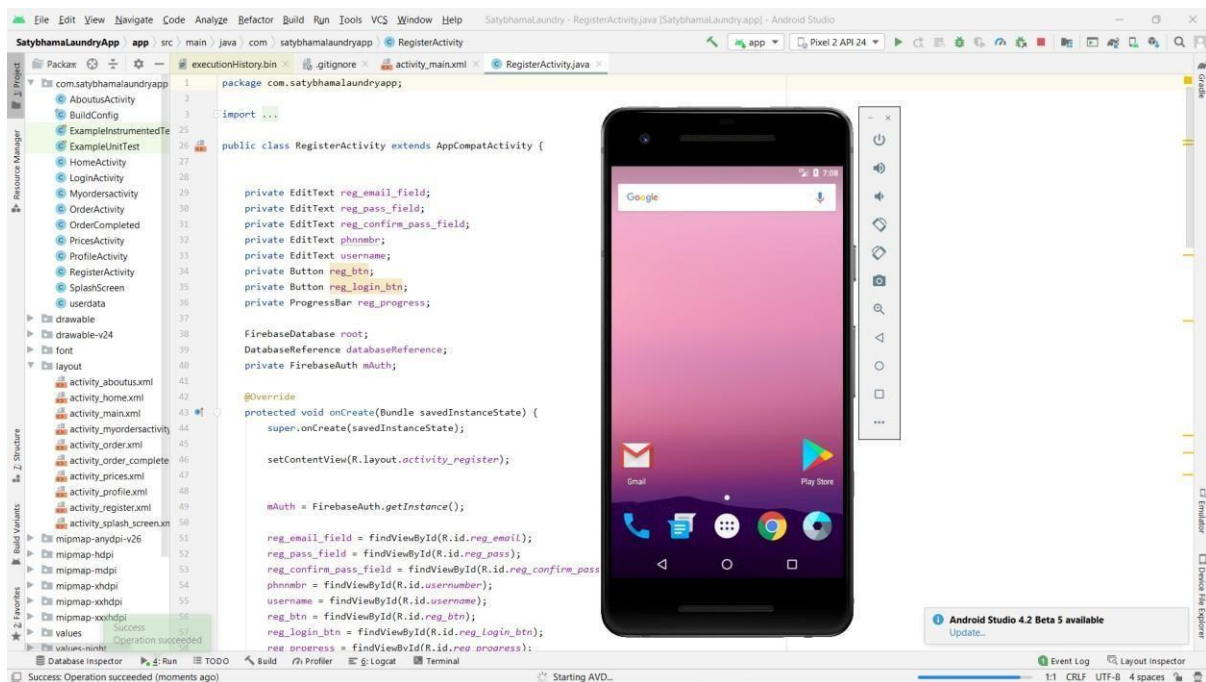
Currently, you can't use the emulator's extended controls when it's running in a tool window. If your development workflow depends heavily on the extended controls, continue to use the Android Emulator as a standalone application. In addition, certain virtual devices—such as Android TV and foldable devices—can't be run in Android Studio because they have specialized UI requirements or important functions in the extended controls.

## CHAPTER 5

### RESULT AND DISCUSSION

#### 5.1: Android studio and emulator

Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug.



**Fig 5.1: Andriod studio and Emulator**

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools.

To support application development within the Android operating system, Android Studio uses a Gradle-based build system, emulator, code templates, and Github integration. Every project in Android Studio has one or more modalities with source code and resource files. These modalities include Android app modules, Library modules, and Google App Engine modules.

Android Studio uses an Instant Push feature to push code and resource changes to a running application. A code editor assists the developer with writing code and offering code completion, refraction, and analysis. Applications built in Android Studio are then compiled into the APK format for submission to the Google Play Store.

The software was first announced at Google I/O in May 2013, and the first stable build was released in December 2014. Android Studio is available for Mac, Windows, and Linux desktop platforms. It replaced Eclipse Android Development Tools (ADT) as the primary IDE for Android application development. Android Studio and the Software Development Kit can be downloaded directly from Google.

### **Emulator:**

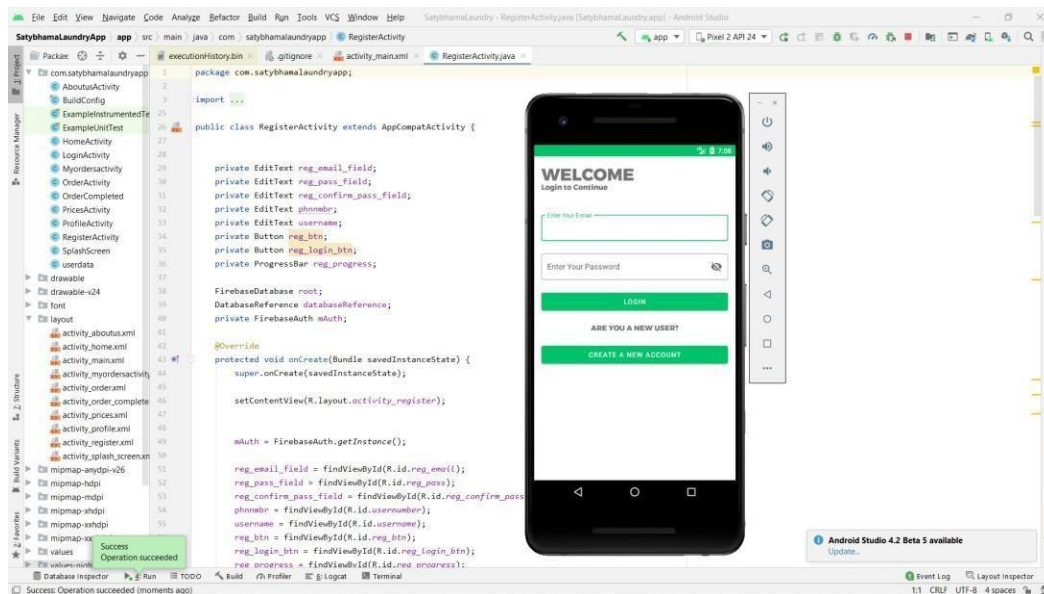
The Android Emulator simulates Android devices on your computer so that you can test your application on a variety of devices and Android API levels without needing to have each physical device.

The emulator provides almost all of the capabilities of a real Android device. You can simulate incoming phone calls and text messages, specify the location of the device, simulate different network speeds, simulate rotation and other hardware sensors, access the Google Play Store, and much more.

Testing your app on the emulator is in some ways faster and easier than doing so on a physical device. For example, you can transfer data faster to the emulator than to a device connected over USB.

The emulator comes with predefined configurations for various Android phone, tablet, Wear OS, and Android TV devices.

## STEP 2:



**Fig 5.2: User Module User Signup**

**5.2 USER MODULE User Signup:** During this section shopper will recruit himself. Once effective info exchange, he/she will register into the framework. Dashboard: During this section, the shopper will see the case along with his vesture solicitation and value of the clothing.

**Laundry Request:** In this segment, client can sent the Vesture demand by filling the Vesture subtlety structure.

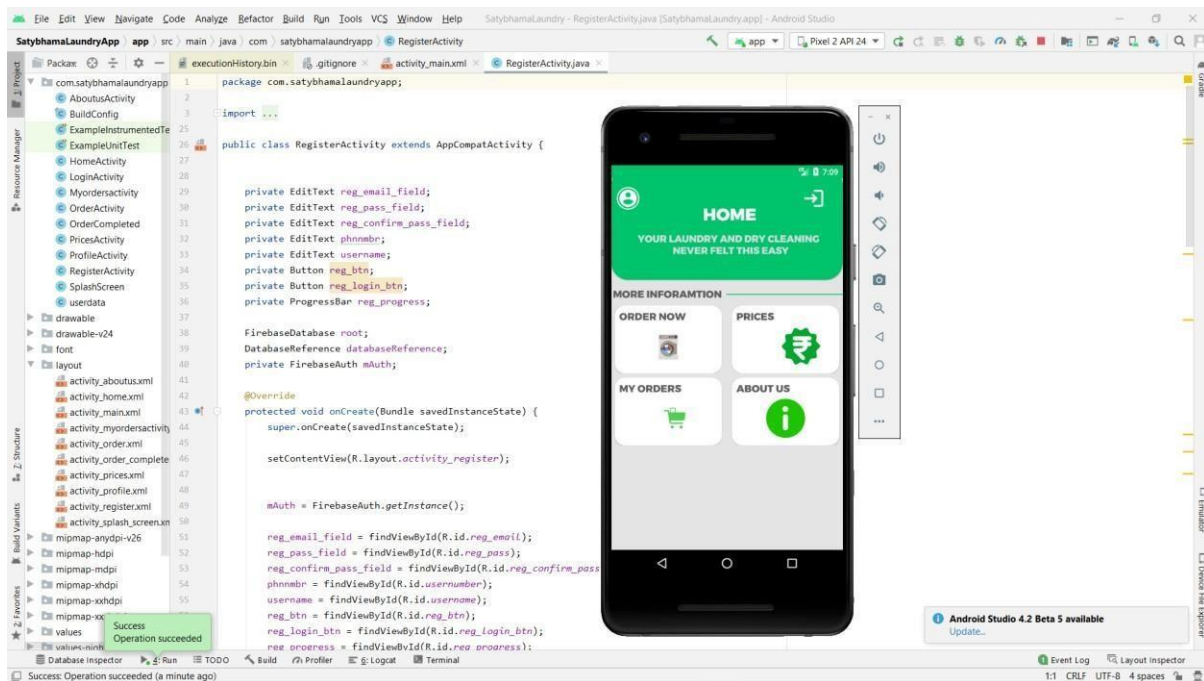
**Request Status:** during this section, shopper will check his/her wear demand standing that is finished by admin (means his/her solicitation is at what step). A shopper will likewise refresh his/her profile, amendment the key and recuperate the key word.

**5.3 ADMIN MODULE Dashboard:** During this section, administrator will see wear solicitation of all user (means the quantity of area unit another solicitation, acknowledge the solicitation, in-measure solicitation and end demand) in an exceedingly shell.

**Reg Users:** During this section, administrator will see shopper detail and update shopper detail. **Laundry Request:** during this half, administrator will see the detail of

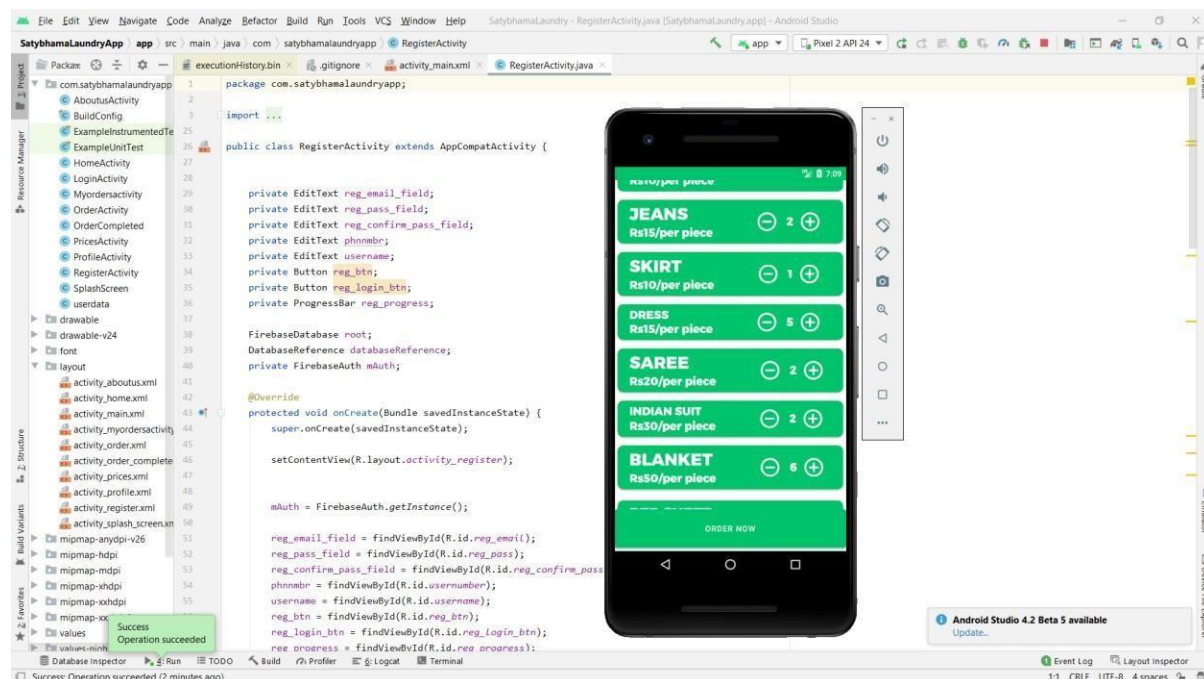
the vesture demand and moreover amendment the part of the clothing demand. Manage

### STEP 3:



**Fig 5.3: Admin Module Dashboard**

### STEP 4:



**Fig 5.4: Orderd items**

## STEP 5: Laundry Price

In this part, administrator can change the Vesture value as indicated by a market cost.

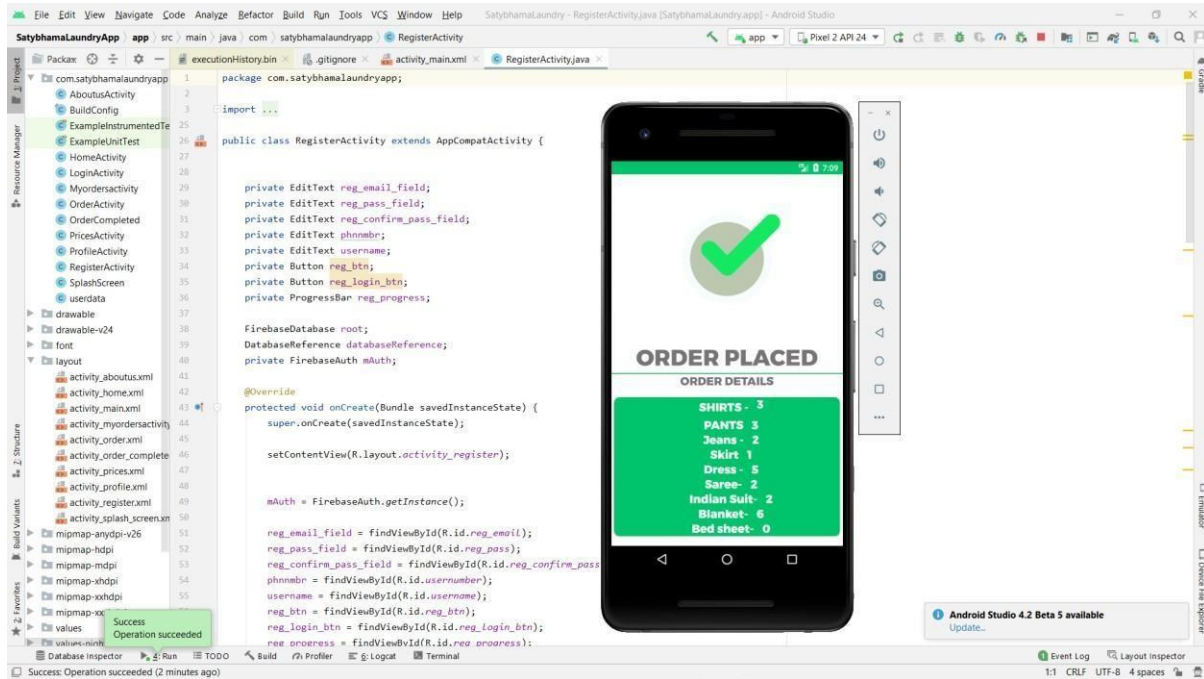


Fig 5.5: Laundry Price

## STEP 6:

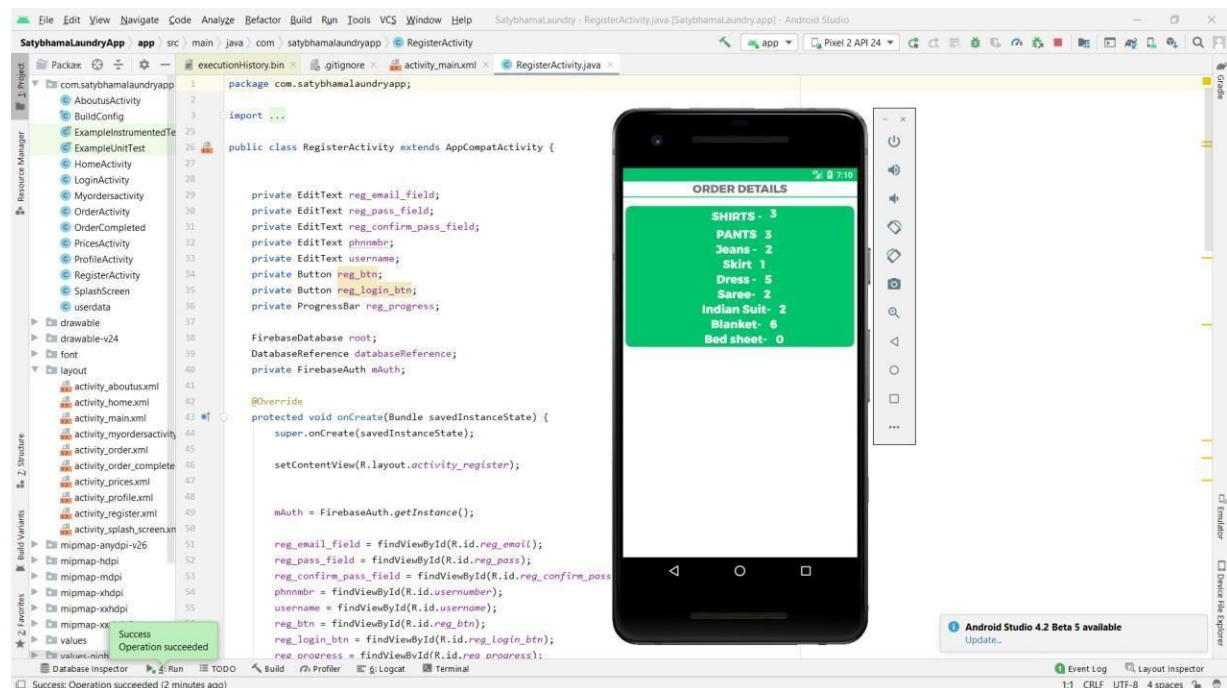


Fig 5.6: Order Details

## STEP 7:

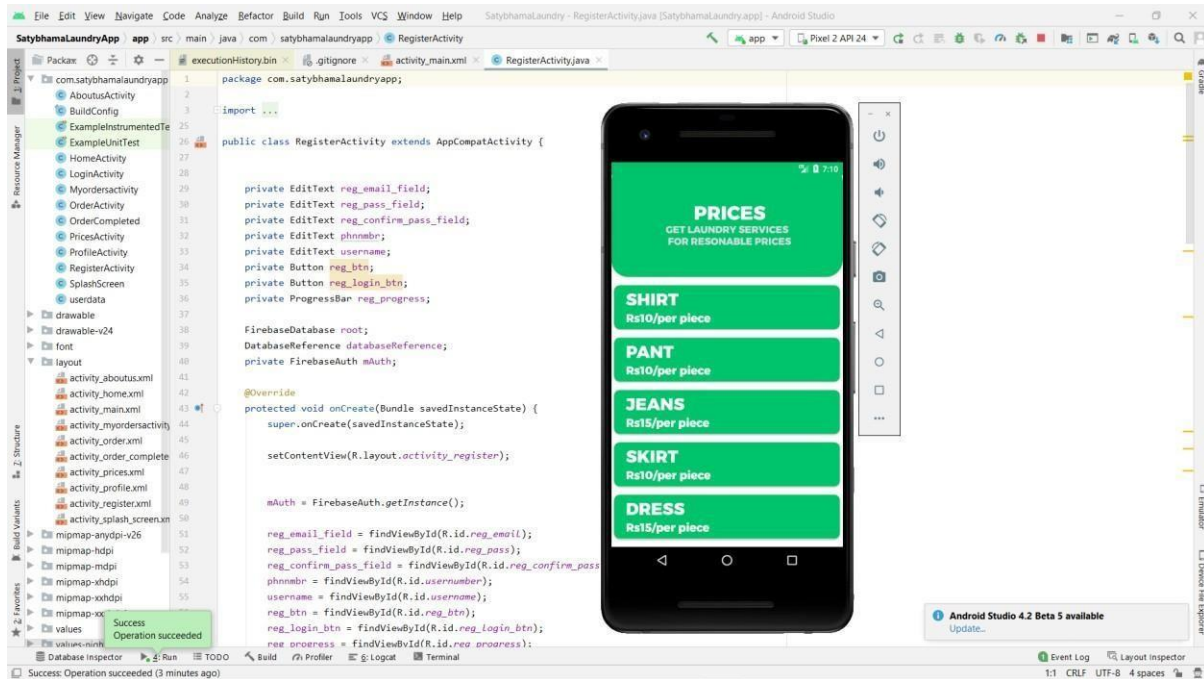


Fig 5.7: prices of ordered items

## STEP 8:

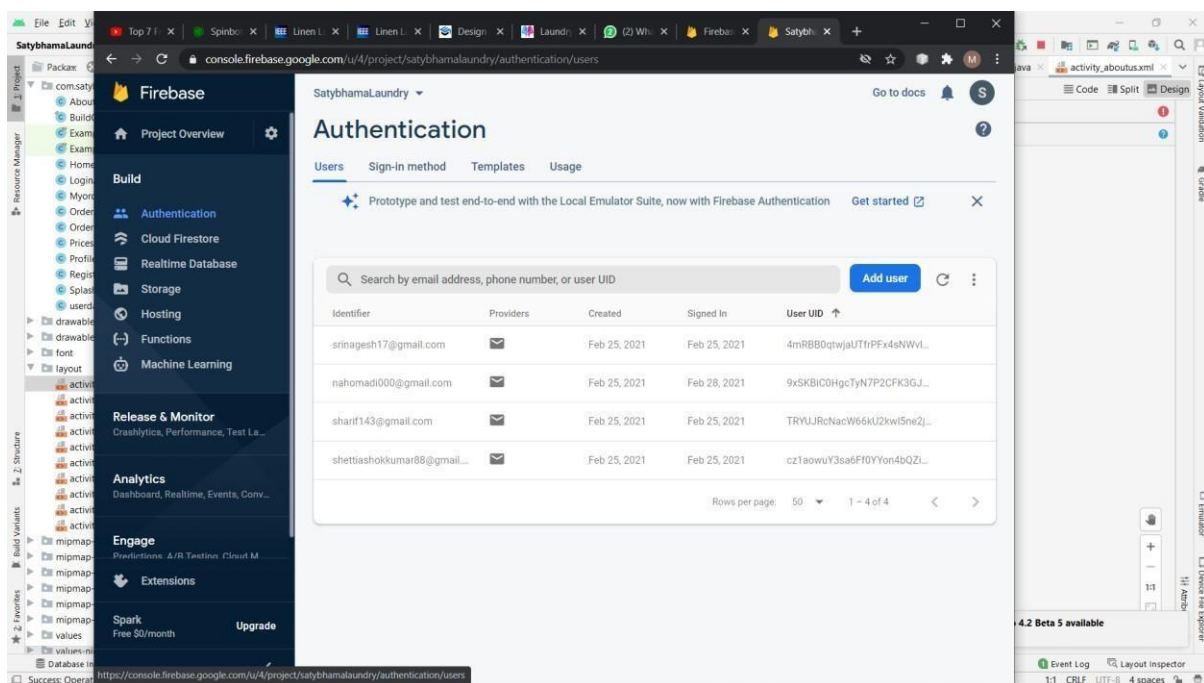
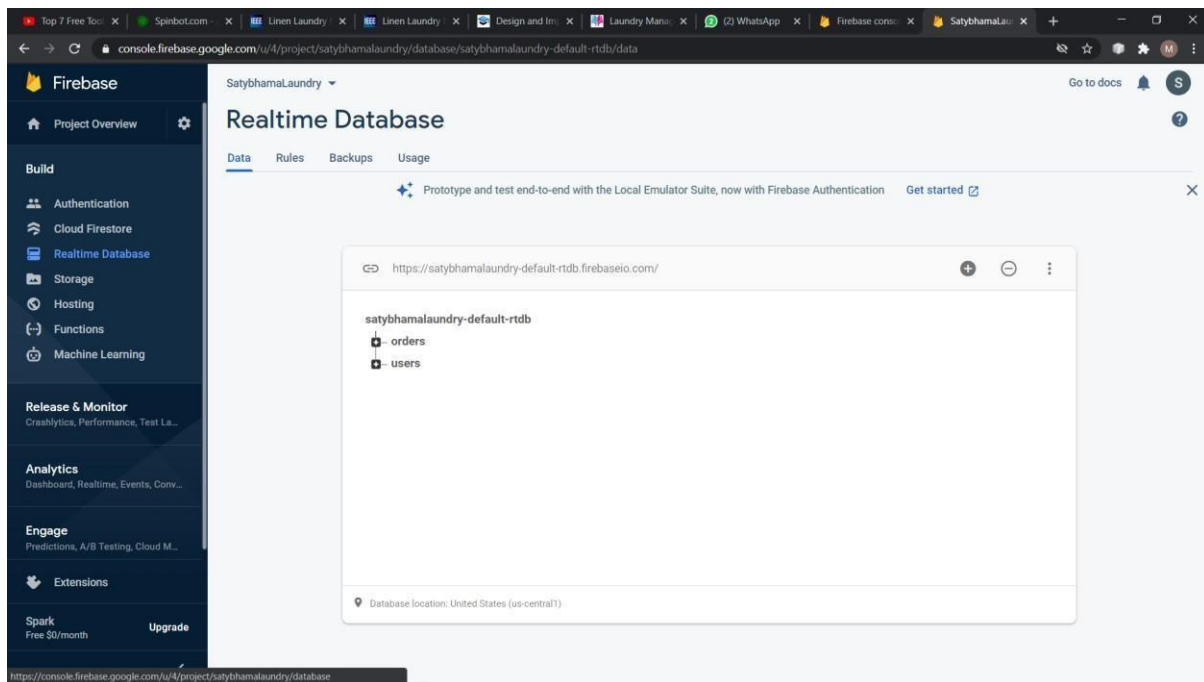


Fig 5.8: Real time data base

## STEP 9:



***Fig5.9: Real time data base***

Real-time Database helps to store and Synchronize Data

Another advantages of a real-time database for the developers is that they do not need the support of backend to build apps as it comes for various platforms,including Andriod, ios and Web.

## **CHAPTER 6**

### **CONCLUSION AND SUMMARY**

#### **6.1 CONCLUSION**

The package was designed in such a way that future modifications can be done easily. The following conclusion can be deduced from the development of the project.

- Automation of the entire system improves the efficiency.
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications.
- Updating of information becomes so easier.
- System security, data security and reliability are the striking features.
- The System has adequate scope for modification in future if it is necessary.

#### **6.2 Summary**

We studied client's requirement and developed an Android Laundry App with customized features to manage laundry services. We paid attention to even the smallest details and Improved user experience, customers can seamlessly place their orders by selecting a flexible laundry store available nearby their location.

On placement of order by customers, the order intimation will be sent to the respective laundry store through host platform. Now, the respective laundry store will send a pick up agent with proforma to collect the laundry items. Every store owner will have a web login privilege, to view the order details placed by customers. st-language: EN-US;mso-bidi-language:AR-SA">laundry services. We paid attention to even the smallest details and Improved user experience, customers can seamlessly place their orders by selecting a flexible laundry store available nearby their location.

Every store will have normal TAT and also provision to opt for express delivery (at an extra charge). While placing order customers have to pick the clothes, no of quantity for service, select the cleaning mode (say dry cleaning or etc.). On completion of order placement, user will be shown the approximate delivery time based on the normal turnaround time of the vendors, provisions to mark express delivery is also available. details and Improved user experience, customers can seamlessly place their orders by selecting a flexible laundry store available nearby their location.

## REFERENCES

- [1]Lawrence, Corbitt, Fisher, and Tidwell. 2000. Internet Commerce: Digital Models for Business. Wiley: Sidney, Australia.
- [2]Phillips, P. 2003. e-Business Strategy.McGraw- Hill: London, UK.
- [3]Whitten, Bentley, and Dittman. 2004. System Analysis and Design Methods (5th ed). New York, NY.
- [4]PerlScriptsJavaScripts.com. 2006. "SQL Tutorial, Database Commands, Beginners Guide, Perl, Program". <http://www.perlscriptsjavascripts.com/tutorials/sql/index.XML>, Accessed 3rd of February, 2006.
- [5]XML Elements. W3schools. Retrieved 16 March, 2015. Journals of science of technology
- [6]CSS Introduction. W3schools. Retrieved 16 March, 2015. [7]XML.com. Retrieved on 16 May, 2012.
- [8]SQL: Project Summary. Black Duck Software. Retrieved 17 September, 2012. Journals of Contemporary Mathematical Science.
- [9]ASP Manual: [www.asp.net](http://www.asp.net)
- [10]ASP for Windows: [asp.net](http://asp.net). Retrieved 29 October, 2013. <https://wikipedia.org/wiki/ASP>

## APPENDIX

### A.SOURCE CODE

#### JAVA Code for register page activity

```
package com.satybhamalaundryapp;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle; import
android.provider.Settings; import
android.text.TextUtils; import
android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class RegisterActivity extends AppCompatActivity {

    private EditText reg_email_field;
    private EditText reg_pass_field;
    private EditText reg_confirm_pass_field;
    private EditText phnnmbr;
    private EditText username;
    private Button reg_btn;
    private Button reg_login_btn;
    private ProgressBar reg_progress;

    FirebaseDatabase root;
    DatabaseReference databaseReference;
    private FirebaseAuth mAuth;

    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_register);

    mAuth = FirebaseAuth.getInstance();

    reg_email_field = findViewById(R.id.reg_email);
    reg_pass_field = findViewById(R.id.reg_pass);
    reg_confirm_pass_field = findViewById(R.id.reg_confirm_pass);
    phnnmbr = findViewById(R.id.usernumber);
    username = findViewById(R.id.username);
    reg_btn = findViewById(R.id.reg_btn);
    reg_login_btn = findViewById(R.id.reg_login_btn);
    reg_progress = findViewById(R.id.reg_progress);

    reg_login_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            //Hides the keyboard
            InputMethodManager inputManager = (InputMethodManager)
                getSystemService(Context.INPUT_METHOD_SERVICE);
            inputManager.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(),
                InputMethodManager.HIDE_NOT_ALWAYS);

            Intent regIntent = new Intent(RegisterActivity.this, LoginActivity.class);
            startActivity(regIntent);

        }
    });

    reg_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            //Hides the keyboard
            InputMethodManager inputManager = (InputMethodManager)
                getSystemService(Context.INPUT_METHOD_SERVICE);
            inputManager.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(),
                InputMethodManager.HIDE_NOT_ALWAYS);
            root = FirebaseDatabase.getInstance("https://satybamalaundry-default-
                rtbd.firebaseio.com/");

            databaseReference = root.getReference("users");

            String email = reg_email_field.getText().toString();

```

```

String pass = reg_pass_field.getText().toString();
String confirm_pass = reg_confirm_pass_field.getText().toString();
String number = phnnmbr.getText().toString();
String usernamef = username.getText().toString();

if (!TextUtils.isEmpty(email) && !TextUtils.isEmpty(pass) &
!TextUtils.isEmpty(confirm_pass)) {
    if (pass.equals(confirm_pass)) {
        reg_progress.setVisibility(View.VISIBLE);
        mAuth.createUserWithEmailAndPassword(email,
        pass).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    String m_androidId = Settings.Secure.getString(getContentResolver(),
                    Settings.Secure.ANDROID_ID);
                    userdata userdata = new userdata(usernamef, number, email);
                    databaseReference.child(m_androidId).setValue(userdata);
                    sendToMain();
                } else {
                    String errorMessage = task.getException().getMessage();
                    showSnackBar(errorMessage);
                }
            }
        });
        reg_progress.setVisibility(View.INVISIBLE);
    }
} else {
    showSnackBar("Password fields doesn't match!");
}
}
});
}

private void showSnackBar(String message) {
    Snackbar.make(findViewById(android.R.id.content), message,
    Snackbar.LENGTH_LONG).show();
}

@Override
protected void onStart() {
    super.onStart();
    FirebaseUser currentUser = mAuth.getCurrentUser();
    if(currentUser != null){
        sendToMain();
    }
}
}

```

```

private void sendToMain() {
Intent mainIntent = new Intent(getApplicationContext(), HomeActivity.class);
    startActivity(mainIntent);
    finish();
}
}

```

## LOGIN PAGE JAVA CODE

```

package com.satybhamalaundryapp;

import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.FirebaseFirestore;

public class LoginActivity extends AppCompatActivity {
private EditText loginEmailText;
private EditText loginPassText;
private Button loginBtn;
private Button loginRegBtn;
private ProgressBar loginProgress;
private FirebaseAuth mAuth;
FirebaseFirestore firebaseFirestore;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

//firebase instance

```

```

mAuth = FirebaseAuth.getInstance();
firebaseFirestore = FirebaseFirestore.getInstance();

loginEmailText = findViewById(R.id.usermail);
loginPassText = findViewById(R.id.userpassword);
loginBtn = findViewById(R.id.loginBtn);
loginRegBtn = findViewById(R.id.loginRegBtn);
loginProgress = findViewById(R.id.login_progress);

//action when register button clicked
loginRegBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent regIntent = new Intent(LoginActivity.this, RegisterActivity.class);
        startActivity(regIntent);
    }
});

//action when login button clicked
loginBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //Hides the keyboard
        InputMethodManager inputManager = (InputMethodManager)
            getSystemService(Context.INPUT_METHOD_SERVICE);
        inputManager.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(),
            InputMethodManager.HIDE_NOT_ALWAYS);
        //storing the entered values to a string
        String loginEmail = loginEmailText.getText().toString();
        String loginPass = loginPassText.getText().toString();

        if (TextUtils.isEmpty(loginEmail) || TextUtils.isEmpty(loginPass)) {
            Toast.makeText(getApplicationContext(), "ENTER Your DETAILS",
                Toast.LENGTH_SHORT);
        }

        if (!TextUtils.isEmpty(loginEmail) && !TextUtils.isEmpty(loginPass)) {
            loginProgress.setVisibility(View.VISIBLE);

            mAuth.signInWithEmailAndPassword(loginEmail,
                loginPass).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        sendToMain();
                    } else {
                        String errorMessage = task.getException().getMessage();
                        showSnackBar(errorMessage);
                    }
                }
            });
            loginProgress.setVisibility(View.INVISIBLE);
        }
    }
});

```

```

        });
    }
}
});
}

private void sendToMain() {
Intent mainIntent = new Intent(getApplicationContext(), HomeActivity.class);
    startActivity(mainIntent);

}

@SuppressLint("WrongConstant")
private void showSnackBar(String message) {
Snackbar.make(findViewById(android.R.id.content), message,
Toast.LENGTH_LONG).show();
}

@Override
protected void onStart() {
super.onStart();

FirebaseUser currentUser = mAuth.getCurrentUser();
if (currentUser != null) {
    sendToMain();
}
}
}

```

## XML CODE FOR REGISTER PAGE AND LOGIN PAGE

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:padding="15dp"
tools:context=".RegisterActivity">
<LinearLayout
android:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="match_parent">

<TextView

android:layout_width="wrap_content"

```

```
android:layout_height="wrap_content"
android:fontFamily="@font/ms"
android:text="WELCOME"
android:textSize="35dp">
```

```
</TextView>
```

```
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="-8dp"
android:fontFamily="@font/ms"
android:text="Create A New Account"
android:textSize="15dp">
```

```
</TextView>
```

```
<ProgressBar
android:id="@+id/reg_progress"
style="?android:attr/progressBarStyleHorizontal"
android:layout_width="200dp"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:layout_marginBottom="10dp"
android:indeterminate="true"
android:visibility="gone" />
```

```
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
android:gravity="center"
android:orientation="vertical">
```

```
<com.google.android.material.textfield.TextInputLayout
```

```
style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
app:endIconMode="clear_text"
android:hint="Enter Your Name">
```

```
<com.google.android.material.textfield.TextInputEditText
android:id="@+id/username"
android:inputType="text"
android:layout_width="match_parent"
android:layout_height="wrap_content">
```

```
</com.google.android.material.textfield.TextInputEditText>
```

```
</com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.material.textfield.TextInputLayout  
  
style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"  
android:layout_width="match_parent"  
app:endIconMode="clear_text"  
android:layout_height="wrap_content"  
android:layout_marginTop="20dp"  
android:hint="Enter Your Mobile Number">
```

```
<com.google.android.material.textfield.TextInputEditText  
android:id="@+id/usernumber"  
android:inputType="number"  
android:layout_width="match_parent"  
android:layout_height="wrap_content">
```

```
</com.google.android.material.textfield.TextInputEditText>  
</com.google.android.material.textfield.TextInputLayout>  
<com.google.android.material.textfield.TextInputLayout
```

```
style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
app:endIconMode="clear_text"  
android:layout_marginTop="20dp"  
android:hint="Enter Your E-mail">
```

```
<com.google.android.material.textfield.TextInputEditText  
android:id="@+id/reg_email"  
android:inputType="textEmailAddress"  
android:layout_width="match_parent"  
android:layout_height="wrap_content">
```

```
</com.google.android.material.textfield.TextInputEditText>  
</com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.material.textfield.TextInputLayout  
app:endIconMode="password_toggle"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"
```

```
android:layout_marginTop="20dp"
```

```
style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"  
android:hint="Enter Your Password">
```

```
<com.google.android.material.textfield.TextInputEditText  
android:id="@+id/reg_pass"
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content">
```

```
</com.google.android.material.textfield.TextInputEditText>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
app:enablePasswordToggle="true"
style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
android:hint="Re-enter Your Password">
```

```
<com.google.android.material.textfield.TextInputEditText
android:id="@+id/reg_confirm_pass"
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content">
```

```
</com.google.android.material.textfield.TextInputEditText>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.button.MaterialButton
android:id="@+id/reg_btn"
android:layout_width="match_parent"
android:layout_height="50dp"
android:gravity="center"
android:layout_marginTop="20dp"
android:text="REGISTER">
</com.google.android.material.button.MaterialButton>
```

```
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
android:fontFamily="@font/ms"
android:text="ALREADY HAVE AN ACCOUNT?"
android:textSize="15dp">
</TextView>
```

```
<com.google.android.material.button.MaterialButton
android:id="@+id/reg_login_btn"
android:layout_width="match_parent"
android:layout_height="50dp"
android:layout_marginTop="20dp"
android:gravity="center"
android:text="Login">
</com.google.android.material.button.MaterialButton>
</LinearLayout>
```

```
</LinearLayout>
</ScrollView>
```

### HOME PAGE JAVA CODE

```
package com.satybhamalaundryapp;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import androidx.appcompat.app.AppCompatActivity;
import com.google.firebase.auth.FirebaseAuth;
import de.hdodenhof.circleimageview.CircleImageView;
public class HomeActivity extends AppCompatActivity {
    RelativeLayout orderp, pricesp, myordersp, team;
    CircleImageView profilep;
    ImageView logout;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        orderp = findViewById(R.id.order);
        pricesp = findViewById(R.id.prices);
        myordersp = findViewById(R.id.morders);
        team = findViewById(R.id.team);
        profilep = findViewById(R.id.profile_image);
        logout = findViewById(R.id.logout);
        logout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FirebaseAuth.getInstance().signOut();
                Intent intent = new Intent(getApplicationContext(), LoginActivity.class);
                startActivity(intent);
                finish();
            }
        });
        profilep.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), ProfileActivity.class);
                startActivity(intent);
                finish();
            }
        });
        orderp.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), OrderActivity.class);
```

```

        startActivity(intent);
        finish();
    }
});

```

*//when user clicks on prices*

```

pricesp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(), PricesActivity.class);
        startActivity(intent);
        finish();
    }
});

```

*//when user clicks on myorder*

```

myordersp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(), Myordersactivity.class);
        startActivity(intent);
        finish();
    }
});

```

*//when user clicks on team*

```

team.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(), AboutusActivity.class);
        startActivity(intent);
        finish();
    }
});

```

```

}

```

*//when user clicks on back app closes*

```

public void quit() {
    Intent start = new Intent(Intent.ACTION_MAIN);
    start.addCategory(Intent.CATEGORY_HOME);
    start.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    start.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(start);
}
@Override
public void onBackPressed() {
    quit();
}

```

```
}  
}
```

## HOME PAGE XML CODE

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
  xmlns:android="http://schemas.android.com/apk/res/android"  
  xmlns:app="http://schemas.android.com/apk/res-auto"  
  xmlns:tools="http://schemas.android.com/tools"  
  android:layout_width="match_parent"  
  android:layout_height="match_parent"  
  android:orientation="vertical"  
  android:background="#E4E4E4"  
  tools:context=".HomeActivity">
```

```
  <androidx.cardview.widget.CardView  
    android:layout_width="match_parent"  
    android:layout_height="200dp"  
    android:padding="10dp"  
    android:layout_marginTop="-20dp"  
    android:paddingLeft="5dp"  
    android:paddingRight="5dp"  
    app:cardBackgroundColor="#05c46b"  
    app:cardCornerRadius="30dp"
```

```
    tools:ignore="MissingConstraints">
```

```
  <TextView  
    android:layout_width="wrap_content"  
    android:layout_height="70dp"  
    android:layout_marginStart="5dp"  
    android:fontFamily="@font/ms"  
    android:layout_gravity="center"  
    android:textColor="#ffffff"  
    android:textSize="30dp"  
    android:text="HOME">
```

```
</TextView>
```

```
<TextView  
  android:layout_width="wrap_content"  
  android:layout_height="70dp"  
  android:layout_marginStart="5dp"  
  android:layout_marginTop="38dp"  
  android:fontFamily="@font/ms"  
  android:layout_gravity="center"  
  android:textColor="#d2dae2"  
  android:textSize="17dp"  
  android:gravity="center"
```

```
android:text="YOUR LAUNDRY AND DRY CLEANING \n NEVER FELT THIS EASY">
```

```
</TextView>
```

```
<de.hdodenhof.circleimageview.CircleImageView
xmlns:app="http://schemas.android.com/apk/res-auto"
android:id="@+id/profile_image"
android:layout_width="40dp"
android:layout_height="50dp"
android:layout_marginTop="30dp"
android:layout_marginStart="10dp"
android:src="@drawable/ic_baseline_account_circle_24"
app:civ_border_width="2dp"
app:civ_border_color="#FFFFFF"/>
```

```
<ImageView
android:id="@+id/logout"
android:layout_width="40dp"
android:layout_height="50dp"
android:layout_marginTop="30dp"
android:layout_marginStart="340dp"
android:src="@drawable/logout" />
```

```
</androidx.cardview.widget.CardView>
```

```
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="14dp"
android:orientation="horizontal">
```

```
<TextView
android:layout_marginStart="3dp"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:fontFamily="@font/ms"
android:text="MORE INFORAMTION"
android:textColor="#5E5E5E"
android:textSize="18dp" />
```

```
<View
android:layout_width="match_parent"
android:layout_height="2dp"
android:layout_gravity="center"
android:layout_marginLeft="8dp"
android:background="@color/purple_700"></View>
```

```
</LinearLayout
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:orientation="horizontal">
```

```
<androidx.cardview.widget.CardView
```

```
    android:layout_marginStart="6dp"
    android:layout_width="160dp"
    android:layout_height="120dp"
    android:layout_weight="1"
    app:cardCornerRadius="18dp"
    app:cardElevation="2dp">
```

```
<RelativeLayout
    android:id="@+id/order"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="8dp">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/ms"
    android:text="ORDER NOW"
    android:textColor="#5E5E5E"
    android:textSize="18dp" />
```

```
<com.airbnb.lottie.LottieAnimationView
```

```
    android:layout_width="wrap_content"
    android:layout_height="80dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    app:lottie_autoPlay="true"
    app:lottie_fileName="anima.json"
    app:lottie_loop="true"
    tools:ignore="MissingConstraints">
```

```
</com.airbnb.lottie.LottieAnimationView>
```

```
</RelativeLayout>
```

```
</androidx.cardview.widget.CardView>
```

```
<androidx.cardview.widget.CardView
    android:layout_width="160dp"
    android:layout_marginRight="6dp"
    android:layout_height="120dp"
```

```
android:layout_marginLeft="18dp"
android:layout_weight="1"
app:cardCornerRadius="18dp"
app:cardElevation="2dp">
```

```
<RelativeLayout
android:id="@+id/prices"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:padding="8dp">
```

```
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:fontFamily="@font/ms"
android:text="PRICES"
android:textColor="#5E5E5E"
android:textSize="18dp" />
```

```
<com.airbnb.lottie.LottieAnimationView
android:layout_width="wrap_content"
android:layout_height="80dp"
android:layout_alignParentRight="true"
android:layout_alignParentBottom="true"
app:lottie_autoPlay="true"
app:lottie_fileName="money.json"
app:lottie_loop="true"
tools:ignore="MissingConstraints">
```

```
</com.airbnb.lottie.LottieAnimationView>
```

```
</RelativeLayout>
```

```
</androidx.cardview.widget.CardView>
```

```
</LinearLayout>
```

```
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="8dp"
android:orientation="horizontal">
```

```
<androidx.cardview.widget.CardView
```

```
android:layout_marginStart="6dp"
android:layout_width="160dp"
android:layout_height="120dp"
```

```
android:layout_weight="1"
app:cardCornerRadius="18dp"
app:cardElevation="2dp">
```

```
<RelativeLayout
android:id="@+id/morders"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:padding="8dp">
```

```
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:fontFamily="@font/ms"
android:text="MY ORDERS"
android:textColor="#5E5E5E"
android:textSize="18dp" />
```

```
<com.airbnb.lottie.LottieAnimationView
android:id="@+id/animation_view"
android:layout_width="wrap_content"
android:layout_height="80dp"
android:layout_alignParentRight="true"
android:layout_alignParentBottom="true"
app:lottie_autoPlay="true"
app:lottie_fileName="cart.json"
app:lottie_loop="true"
tools:ignore="MissingConstraints">
```

```
</com.airbnb.lottie.LottieAnimationView>
```

```
</RelativeLayout>
```

```
</androidx.cardview.widget.CardView>
```

```
<androidx.cardview.widget.CardView
android:layout_width="160dp"
android:layout_marginRight="6dp"
android:layout_height="120dp"
android:layout_marginLeft="18dp"
android:layout_weight="1"
app:cardCornerRadius="18dp"
app:cardElevation="2dp">
```

```
<RelativeLayout
android:id="@+id/team"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:padding="8dp">
```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/ms"
    android:text="ABOUT US"
    android:textColor="#5E5E5E"
    android:textSize="18dp" />

<com.airbnb.lottie.LottieAnimationView
    android:layout_width="wrap_content"
    android:layout_height="80dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    app:lottie_autoPlay="true"
    app:lottie_fileName="about.json"
    app:lottie_loop="true"
    tools:ignore="MissingConstraints"></com.airbnb.lottie.LottieAnimationView>

</RelativeLayout>

</androidx.cardview.widget.CardView>

</LinearLayout>

</LinearLayout>

```

### **ABOUTUS JAVA CODE**

```

package com.satybhamalaundryapp;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

public class AboutusActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_aboutus);
    }
}

```

### **MY ORDERS JAVA CODE**

```

package com.satybhamalaundryapp;

import android.os.Bundle; import
android.provider.Settings; import
android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
import com.firebase.client.DataSnapshot;
import com.firebase.client.Firebase;
import com.firebase.client.FirebaseError;
import com.firebase.client.ValueEventListener;
public class Myordersactivity extends AppCompatActivity {
    Firebase myfirebase;
    TextView shirtfcounTF, pantsfcounTF, jeansfcounTF, SkirtfcounTF, dressfcounTF,
    SareefcounTF, indsfcounTF, blanketfcounTF, bedsheetfcounTF;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_myordersactivity);
        shirtfcounTF = findViewById(R.id.shirtfcounTF);
        pantsfcounTF = findViewById(R.id.pantsfcounTF);
        jeansfcounTF = findViewById(R.id.jeansfcounTF);
        SkirtfcounTF = findViewById(R.id.SkirtfcounTF);
        dressfcounTF = findViewById(R.id.dressfcounTF);
        SareefcounTF = findViewById(R.id.SareefcounTF);
        indsfcounTF = findViewById(R.id.indsfcounTF);
        blanketfcounTF = findViewById(R.id.blanketfcounTF);
        bedsheetfcounTF = findViewById(R.id.bedsheetfcounTF);

        String shirtfcounTF = getIntent().getStringExtra("key1");
        String pantsfcounTF = getIntent().getStringExtra("key2");
        String jeansfcounTF = getIntent().getStringExtra("key3");
        String SkirtfcounTF = getIntent().getStringExtra("key4");
        String dressfcounTF = getIntent().getStringExtra("key5");
        String SareefcounTF = getIntent().getStringExtra("key6");
        String indsfcounTF = getIntent().getStringExtra("key7");
        String blanketfcounTF = getIntent().getStringExtra("key8");
        String bedsheetfcounTF = getIntent().getStringExtra("key9");

        shirtfcounTF.setText(shirtfcounTF);
        pantsfcounTF.setText(pantsfcounTF);
        jeansfcounTF.setText(jeansfcounTF);
        SkirtfcounTF.setText(SkirtfcounTF);
        dressfcounTF.setText(dressfcounTF);
        SareefcounTF.setText(SareefcounTF);
        indsfcounTF.setText(indsfcounTF);
    }
}

```

```
blanketfcoun.setText(blanketfcoun);
bedsheetfcoun.setText(bedsheetfcoun);
```

```

Firebase.setAndroidContext(this);
String m_androidId = Settings.Secure.getString(getContentResolver(),
Settings.Secure.ANDROID_ID);
myfirebase = new Firebase("https://satybhamalaundry-default-
rtbd.firebaseio.com/orders/"+m_androidId);
myfirebase.addValueEventListener(new ValueEventListener() {
@Override
public void onDataChange(DataSnapshot dataSnapshot) {

String shirtfcoun = dataSnapshot.child("shirts").getValue(String.class);
String pantsfcoun = dataSnapshot.child("pants").getValue(String.class);
String jeansfcoun = dataSnapshot.child("jeans").getValue(String.class);
String Skirtfcoun = dataSnapshot.child("skirt").getValue(String.class);
String dressfcoun = dataSnapshot.child("dress").getValue(String.class);
String Sareefcoun = dataSnapshot.child("saree").getValue(String.class);
String indsfoun = dataSnapshot.child("indian suuit").getValue(String.class);
String blanketfcoun = dataSnapshot.child("blanket").getValue(String.class);
String bedsheetfcoun = dataSnapshot.child("bed sheet").getValue(String.class);
```

```

shirtcount.setText(shirtfcoun);
pantscount.setText(pantsfcoun);
jeanscount.setText(jeansfcoun);
Skirtcount.setText(Skirtfcoun);
dresscount.setText(dressfcoun);
Sareefcount.setText(Sareefcoun);
indsfcount.setText(indsfcoun);
blanketcount.setText(blanketfcoun);
bedsheetcount.setText(bedsheetfcoun);
```

```

}
```

```

@Override
public void onCancelled(FirebaseError firebaseError) {
shirtcount.setText("NOT FOUND");
}
});
}
}
```

## ABOUT US XML CODE

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".AboutusActivity">
```

```
<com.airbnb.lottie.LottieAnimationView
android:id="@+id/lottieAnimationView2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignParentRight="true"
android:layout_alignParentBottom="true"
app:lottie_autoPlay="true"
app:lottie_fileName="about.json"
app:lottie_loop="true"
tools:ignore="MissingConstraints">
```

```
</com.airbnb.lottie.LottieAnimationView>
```

```
<View
android:id="@+id/view"
android:layout_width="match_parent"
android:layout_height="2dp"
android:layout_gravity="center"
android:background="@color/purple_700"
app:layout_constraintTop_toBottomOf="@+id/lottieAnimationView2"
tools:layout_editor_absoluteX="16dp">
```

```
</View>
```

```
<TextView
android:id="@+id/textView3"
android:layout_width="match_parent"
android:layout_height="40sp"
android:fontFamily="@font/ms"
android:gravity="center"
android:text="ABOUT US"
android:textSize="40dp"
app:layout_constraintTop_toBottomOf="@+id/view"
tools:ignore="MissingConstraints"
tools:layout_editor_absoluteX="27dp">
```

```
</TextView>
```

```
<TextView
android:id="@+id/textView4"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="36dp"
android:fontFamily="@font/ms"
```

```

android:gravity="center"
android:text=" we are building India's first organized chain of laundry we enable the
time crunched customers to avail pick-n- drop service from the comforts of their
home or office. "
android:textSize="17dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView3"
app:layout_constraintVertical_bias="0.0"
tools:ignore="MissingConstraints"
tools:layout_editor_absoluteX="0dp">

</TextView>

<View

android:layout_width="match_parent"
android:layout_height="2dp"
android:layout_gravity="center"
android:background="@color/purple_700"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView4"
" app:layout_constraintVertical_bias="0.051"
tools:layout_editor_absoluteX="-16dp">

</View>

</androidx.constraintlayout.widget.ConstraintLayout>

```

### **ORDER ACTIVITY JAVA CODE**

```

package com.satybhamalaundryapp;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;
import android.provider.Settings;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;

import com.google.android.material.button.MaterialButton;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class OrderActivity extends AppCompatActivity {

@Override

```

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_order);  
    ImageButton shirtbtnp, shirtbtnm,  
    subpant, addpant,  
    subjeans, addjeans,  
    subskirt, addskirt,  
    address, subdress,  
    addsaree, subsaree,  
    subindsuit, addindsuit,  
    subblanket, addblanket,  
    subbedsheet, addbedsheet;
```

```
    TextView shirtsc,  
    pantscountc,  
    jeanscountc,  
    skirtcountc,  
    dresscountc,  
    sareecountc,  
    indsuitcountc,  
    blanketcountc,  
    bedsheetcountc;
```

```
    final int[] value = {0};  
    final int[] vv = {0};  
    final int[] jeansvalue = {0};  
    final int[] skirtvalue = {0};  
    final int[] dressvalue = {0};  
    final int[] sareevalue = {0};  
    final int[] indsuitvalue = {0};  
    final int[] blanketvalue = {0};  
    final int[] bedshhetvalue = {0};
```

```
    MaterialButton button;  
    final FirebaseDatabase[] root = new FirebaseDatabase[1];  
    final DatabaseReference[] databaseReference = new DatabaseReference[1];
```

```
    shirtbtnp = findViewById(R.id.addshirt);  
    shirtbtnm = findViewById(R.id.subshirt);  
    shirtsc = findViewById(R.id.shirtscout);
```

```
    subpant = findViewById(R.id.subpant);  
    addpant = findViewById(R.id.addpant);  
    pantscountc = findViewById(R.id.pantcount);
```

```
    subjeans = findViewById(R.id.subjeans);  
    addjeans = findViewById(R.id.addjeans);  
    jeanscountc = findViewById(R.id.jeanscount);
```

```
addskirt = findViewById(R.id.addskirt);
subskirt = findViewById(R.id.subskirt);
skirtcountc = findViewById(R.id.skirtcount);
```

```
address = findViewById(R.id.address);
subdress = findViewById(R.id.subdress);
dresscountc = findViewById(R.id.dresscount);
```

```
addsaree = findViewById(R.id.addsaree);
subsaree = findViewById(R.id.subsaree);
sareecountc = findViewById(R.id.sareecount);
```

```
subindsuit = findViewById(R.id.subindsuit);
addindsuit = findViewById(R.id.addindsuit);
indsuitcountc = findViewById(R.id.indsuitcount);
```

```
subblanket = findViewById(R.id.subblanket);
addblanket = findViewById(R.id.addblanket);
blanketcountc = findViewById(R.id.blanketcount);
```

```
subbedsheet = findViewById(R.id.subbedsheet);
addbedsheet = findViewById(R.id.addbedsheet);
bedsheetcountc = findViewById(R.id.bedsheetcount);
```

```
button = findViewById(R.id.send);
```

```
shirtbtnp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        value[0]++;
        shirtsc.setText("" + value[0]);
    }
});
shirtbtnm.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (value[0] == 0) {
            value[0] = 0;
        } else {
            value[0]--;
            shirtsc.setText("" + value[0]);
        }
    }
});
```

```

    });

    addpant.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            vv[0]++;
            pantscountc.setText("" + vv[0]);

        }
    });

    subpant.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (vv[0] == 0) {
                vv[0] = 0;
            } else {
                vv[0]--;
                pantscountc.setText("" + vv[0]);
            }

        }
    });

    addjeans.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            jeansvalue[0]++;
            jeanscountc.setText("" + jeansvalue[0]);

        }
    });

    subjeans.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (jeansvalue[0] == 0) {
                jeansvalue[0] = 0;
            } else {
                jeansvalue[0]--;
                jeanscountc.setText("" + jeansvalue[0]);
            }

        }
    });

    addskirt.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

```

```
skirtvalue[0]++;  
skirtcountc.setText("" + skirtvalue[0]);  
  
    }  
});
```

```
subskirt.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if (skirtvalue[0] == 0) {  
            skirtvalue[0] = 0;  
        } else {  
            skirtvalue[0]--;  
            skirtcountc.setText("" + skirtvalue[0]);  
        }  
    }  
});
```

```
address.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        dressvalue[0]++;  
        dresscountc.setText("" + dressvalue[0]);  
    }  
});
```

```
subdress.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if (dressvalue[0] == 0) {  
            dressvalue[0] = 0;  
        } else {  
            dressvalue[0]--;  
            dresscountc.setText("" + dressvalue[0]);  
        }  
    }  
});
```

```
addsaree.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        sareevalue[0]++;  
        sareecountc.setText("" + sareevalue[0]);  
    }  
});
```

```

    });

    subsaree.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    if (sareevalue[0] == 0) {
    sareevalue[0] = 0;
        } else {
    sareevalue[0]--;
    sareecountc.setText("" + sareevalue[0]);
        }

    }
    });

    addindsuit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    indsuitvalue[0]++;
    indsuitcountc.setText("" + indsuitvalue[0]);

    }
    });

    subindsuit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    if (indsuitvalue[0] == 0) {
    indsuitvalue[0] = 0;
        } else {
    indsuitvalue[0]--;
    indsuitcountc.setText("" + indsuitvalue[0]);
        }

    }
    });

    addblanket.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    blanketvalue[0]++;
    blanketcountc.setText("" + blanketvalue[0]);

    }
    });

    subblanket.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    if (blanketvalue[0] == 0) {

```

```

blanketvalue[0] = 0;
        } else {
blanketvalue[0]--;
blanketcountc.setText("" + blanketvalue[0]);
        }

    }
});
addbedsheet.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
bedshhetvalue[0]++;
bedsheetcountc.setText("" + bedshhetvalue[0]);

    }
});

subbedsheet.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (bedshhetvalue[0] == 0) {
bedshhetvalue[0] = 0;
        } else {
bedshhetvalue[0]--;
bedsheetcountc.setText("" + bedshhetvalue[0]);
        }

    }
});

FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
if (user != null) {
// Name, email address, and profile photo Url

String email = user.getEmail();

// Check if user's email is verified
boolean emailVerified = user.isEmailVerified();

// The user's ID, unique to the Firebase project. Do NOT use this value to
// authenticate with your backend server, if you have one. Use
// FirebaseAuth.getIdToken() instead.
String uid = user.getId();
    }

button.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View v) {
    root[0] = FirebaseDatabase.getInstance("https://satybbhamalaundry-default-
    rtbd.firebaseio.com/");
    String d1 = Settings.Secure.getString(getContentResolver(),
    Settings.Secure.ANDROID_ID);
    //      Date todaysdate = new Date();
    //      SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
    //      String date = format.format(todaysdate);

    databaseReference[0] = root[0].getReference("orders");
    String fshirt = shirtsc.getText().toString();
    String fpant = pantscountc.getText().toString();
    String fjeans = jeanscountc.getText().toString();
    String fskirt = skirtcountc.getText().toString();
    String fsaree = sareecountc.getText().toString();
    String fdress = dresscountc.getText().toString();
    String findcout = indsuitcountc.getText().toString();
    String fblanket = blanketcountc.getText().toString();
    String fbedsheet = bedsheetcountc.getText().toString();

    databaseReference[0].child(d1).child("shirts").setValue(fshirt);
    databaseReference[0].child(d1).child("pants").setValue(fpant);
    databaseReference[0].child(d1).child("jeans").setValue(fjeans);
    databaseReference[0].child(d1).child("skirt").setValue(fskirt);
    databaseReference[0].child(d1).child("dress").setValue(fdress);
    databaseReference[0].child(d1).child("saree").setValue(fsaree);
    databaseReference[0].child(d1).child("indian suuit").setValue(findcout);
    databaseReference[0].child(d1).child("blanket").setValue(fblanket);
    databaseReference[0].child(d1).child("bed sheet").setValue(fbedsheet);

    Intent intent = new Intent(getApplicationContext(), OrderCompleted.class);
    intent.putExtra("key1", fshirt).putExtra("key2", fpant).putExtra("key3", fjeans)
        .putExtra("key4", fskirt).putExtra("key5", fdress).putExtra("key6", fsaree)

        .putExtra("key7", findcout).putExtra("key8", fblanket).putExtra("key9", fbedsheet);

    startActivity(intent);

    }
    });

}

}

```