PARAMEKKAVU COLLEGE OF ARTS AND SCIENCE

THRISSUR, KERALA

(AFFILIATED TO CALICUT UNIVERSITY)

DEPARTMENT OF COMPUTER SCIENCE,

Autism Spectrum Disorder (ASD)

PROJECT REPORT

Submitted in partial fulfilment of the requirements of the award of

BSC COMPUTER SCIENCE

Under

University of Calicut

Submitted by

AISWARYA JOBE

ANOOP K

ATHIRA K R

Under the guidance of

Mrs. SREEDEVI.V

(Department of Computer Science)

PARAMEKKAVU COLLEGE OF ARTS AND SCIENCE

Thrissur, Kerala,680002

(AFFILIATED TO CALICUT UNIVERSITY)

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certified that the project entitled "Autism Spectrum Disorder (ASD)" is a bonafide work done by AISWARYA JOBE, ANOOP K, ATHIRA K R, *Submitted in partial fulfilment of the requirements of the award of* ***BSC COMPUTER SCIENCE ,*** *under my guidance at Paramekkavu College of Arts and Science, Thrissur.*

Place: Thrissur

Reg No:PRAUSCS018,PRAUSCS007,PRAUSCS003

Date:

HOD: Mrs, Saritha E

Guide: Mrs, Sreedevi.V

Submitted for the viva-voce held at PARAMEKKAVU COLLEGE OF ARTS AND SCIENCE, Thrissur on .....

Internal Examiner:

External Examiner:

ACKNOWLEDGEMENT

I am grateful to each and every one who has contributed in numerous ways to the development of this project whose effort had been a helping hand in every footstep towards my destiny.

I gave all the glory and honor to the god almighty, whose divine touch had been the most inspiration and guiding light in the fulfilment of this uphill task.

I am glad to present the project report entitled "Autism spectrum disorder (ASD)" prepared as a part of our sixth semester Bsc Computer Science course.

I would take this opportunity to express my deep sense of gratitude and sincere thanks to our principal **DR. K P SUDHA**, of Paramekkavu College of Arts and Science, Thrissur for giving me an opportunity to take up the project work.

I would like to express my gratitude to **Mrs. SARITHA E** head of the department of Computer science, who has served as a host of valuable corrections. It is wordless to say thanks to my internal guide **Mrs. SREEDEVI V,** and also extend my sincere gratitude to all other Assistant Professors in the Computer Science Department, who have contributed to complete my project.

I extremely thank my parents for their moral support and encouragement, which had always been a source of inspiration to me. Above all I would like to thank the almighty for his abundant blessing, without which nothing is possible.

CONTENTS

INTRODUCTION

INTRODUCTION

Autism Spectrum Disorder (ASD) is a neurological disorder which might have a lifelong impact on the language learning, speech, cognitive, and social skills of an individual. Its symptoms usually show up in the developmental stages, i.e., within the first two years after birth, and it impacts around 1% of the population globally. Autism Spectrum Disorder occurs in the developmental stages of an individual and is a serious disorder which can impair the ability to interact or communicate with others. So, this app will be helpful for students who are autism patients. Student or parent, teacher and admin are the modules. There will be separate login for user and teacher. Admin is the overall controller of the system. Teacher can add students. There will be a set of questions to determine the disorder. After login there will be two sections entertainment and learning section. We can watch songs and stories added by admin in entertainment section. In learning section, student can learn colours and if answer correctly to the students, he/she can earn points. Text to speech technology is used here for teaching. There will be a repeating option here. Like this alphabet, fruits, vegetables can learn. Quiz section, puzzles (image ordering, missing number) etc. also added. All these data are added by admin. A chatbot also assists for doubts and queries. Admin will add doctors and student or teacher can view their contact numbers and consult them. Teacher can conduct and valuate exam according to student's skill. Online counselling will be provided and a payment option also there. Admin can assign a teacher for each set of students. Face recognition technology also added to detect the mood of student (hyper active, sad etc.) and thus decide the chance of autism.

SYSTEM CONFIGURATION

SYSTEM CONFIGURATION

The selection of hardware is important in the existence and proper working of any software. The objective of the system configuration phase is the establishment of the requirement for the system to be acquired, developed, and installed. Analysis of the project is to understand the intricacy and forms the vital part of the system study. Problematic areas are identified, and information is collected. Fact finding or gathering is essential to any analysis of the requirement. It is necessary that the analyst familiarize with objects, activities, and the functions of the device in which the system is implemented.

SOFTWARE REQUIREMENTS

The software configuration says about the software's and packages used to design the system. The following specifications includes the software and packages used by the programmer.

- Platform: Android Studio
- Front end: Android IDE (Java code)
- Language: Java
- Back end: PHP, MySQL
- Operating system: Microsoft Windows 10

HARDWARE REQUIREMENTS

- PROCESSOR: PENTIUM 13
- RAM: 4 GB
- HARD DISK: 1 TB

SYSTEM ANALYSIS

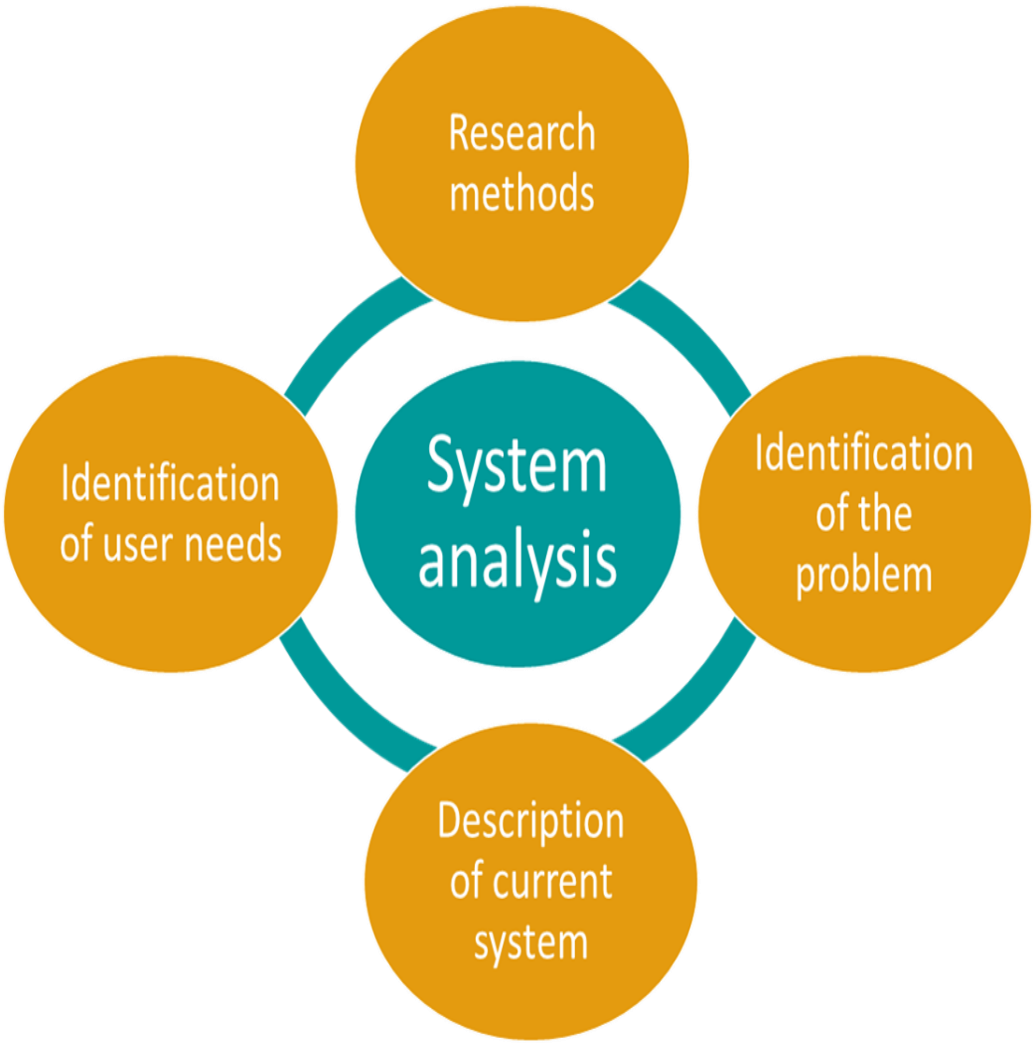
SYSTEM ANALYSIS

System analysis in autism refers to the examination of the complex systems that interact with and influence individuals with autism spectrum disorder (ASD). This includes examining the ways in which social, cultural, and environmental factors impact the experiences and behaviors of individuals with ASD, as well as how these individuals navigate and interact with these systems.

One example of system analysis in autism is examining the ways in which social systems, such as schools or workplaces, can be designed to better accommodate individuals with ASD. This may include providing accommodations such as visual schedules, quiet spaces, or flexible work arrangements. System analysis may also involve examining the ways in which medical systems, such as hospitals or clinics, can provide effective and accessible care for individuals with ASD.

Another important aspect of system analysis in autism is examining the role of families and caregivers in supporting individuals with ASD. This may involve examining the ways in which families can navigate healthcare and educational systems to ensure their loved ones receive the support they need, as well as identifying resources and strategies that can help families cope with the challenges of raising a child with ASD.

Overall, system analysis in autism is a critical tool for understanding the complex and varied experiences of individuals with ASD and identifying opportunities for improvement in the systems that impact their lives.


Research methods
Identification of user needs
System analysis
Identification of the problem
Description of current system

a.EXISTING SYSTEM

Diagnostic and Statistical Manual of Mental Disorders (DSM): The DSM is a widely used diagnostic manual for mental health conditions, including autism. The DSM-5 criteria for autism include difficulties in social communication and interaction, restricted and repetitive behaviors, and sensory processing issues.

1.Autism Diagnostic Observation Schedule (ADOS): The ADOS is a standardized assessment tool used by clinicians to evaluate social communication, social interaction, and play skills in individuals with autism. It involves structured play activities and observations of the individual's behavior.

2.Applied Behavior Analysis (ABA): ABA is an evidence-based therapy used to treat autism. It involves breaking down complex skills into smaller, more manageable steps and using positive reinforcement to teach new skills.

3.Speech Therapy: Many individuals with autism have difficulties with language and communication. Speech therapy can help improve communication skills, including speech and language comprehension, social communication, and pragmatic language.

4.Occupational Therapy: Occupational therapy can help individuals with autism develop daily living skills, such as dressing, feeding, and toileting. It can also address sensory processing issues, motor coordination, and fine motor skills.

5.Social Skills Training: Social skills training involves teaching individuals with autism appropriate social behavior and communication skills, including turn-taking, initiating and maintaining conversations, and nonverbal communication.

These are just a few examples of the existing systems and approaches used in the diagnosis, assessment, and treatment of autism. Treatment plans are often individualized and may involve a combination of different therapies and interventions.

b. PROPOSED SYSTEM

There are several proposed systems and approaches in the diagnosis, assessment, and treatment of autism, some of which are still being researched or in development. Here are a few examples:

1.Early Intervention: Research has shown that early intervention for autism can improve outcomes and increase the likelihood of positive outcomes. Proposed systems and approaches in early intervention include parent-mediated interventions, where parents are trained to provide intervention strategies at home, and early screening programs to identify children at risk for autism at an early age.

2.Technology-Based Interventions: Technology-based interventions, such as mobile applications, virtual reality, and robotics, have shown promise in providing personalized and engaging interventions for individuals with autism. These interventions can target specific skills, such as social communication, language, and behavior.

3.Cognitive Behavioral Therapy (CBT): CBT is a form of psychotherapy that focuses on changing negative thoughts and behaviors. It has shown promise in treating anxiety and depression in individuals with autism.

4.Mindfulness-Based Interventions: Mindfulness-based interventions, such as mindfulness meditation and yoga, have shown promise in reducing stress and improving social communication and behavior in individuals with autism.

5.Personalized Medicine: Personalized medicine involves tailoring medical treatment to an individual's unique genetic and environmental factors. Proposed systems and approaches impersonalized medicine for autism include identifying genetic markers for autism and developing individualized treatment plans based on an individual's genetic profile.

These are just a few examples of proposed systems and approaches in the diagnosis, assessment, and treatment of autism. Further research and development are needed to determine the efficacy and feasibility of these interventions.

C.FEASIBILITY STUDY

❖ Resource requirements

This would involve assessing the financial, human, and material resources required for the intervention or program. This would include costs associated with training staff, purchasing equipment, and maintaining the program.

❖ Effectiveness

The feasibility study would evaluate the effectiveness of the proposed intervention or program in improving the quality of life and functional outcomes for individuals with ASD. This may involve measuring changes in social skills, communication abilities, and independence.

❖ Sustainability

The study would assess the feasibility of sustaining the intervention or program over time, including its long-term impact and the availability of resources to continue the program.

❖ Ethical considerations

Feasibility studies should consider the ethical implications of any proposed intervention or program, including potential risks and benefits, as well as any potential conflicts of interest.

- **Stakeholder input**

  Feasibility studies should involve input from individuals on the autism spectrum, their families, and other stakeholders, to ensure that the proposed intervention or program is meeting their needs and expectations.

A feasibility study in ASD would involve a thorough analysis of the proposed intervention or program to determine whether it is practical, effective, and sustainable in improving outcomes for individuals with ASD.

SYSTEM DESIGN & DEVELOPMENT

SYSTEM DESIGN & DEVELOPMENT

System design and development can play an important role in addressing the needs of individuals with autism. Autism is a developmental disorder that affects communication, social interaction, and behavior. As a result, individuals with autism often require specialized support and accommodations to help them navigate the world around them.

One area where system design and development can be particularly helpful for individuals with autism is in the creation of assistive technology. Assistive technology refers to devices, tools, and software that are designed to help individuals with disabilities perform tasks and activities that might be challenging without additional support. For individuals with autism, this might include communication devices, sensory support tools, or social skills training software.

In addition to assistive technology, system design and development can also be used to create systems and processes that support individuals with autism in various settings. For example, in educational settings, system design and development could be used to create personalized learning plans for students with autism, or to develop online resources and support systems that can be accessed by parents and teachers.

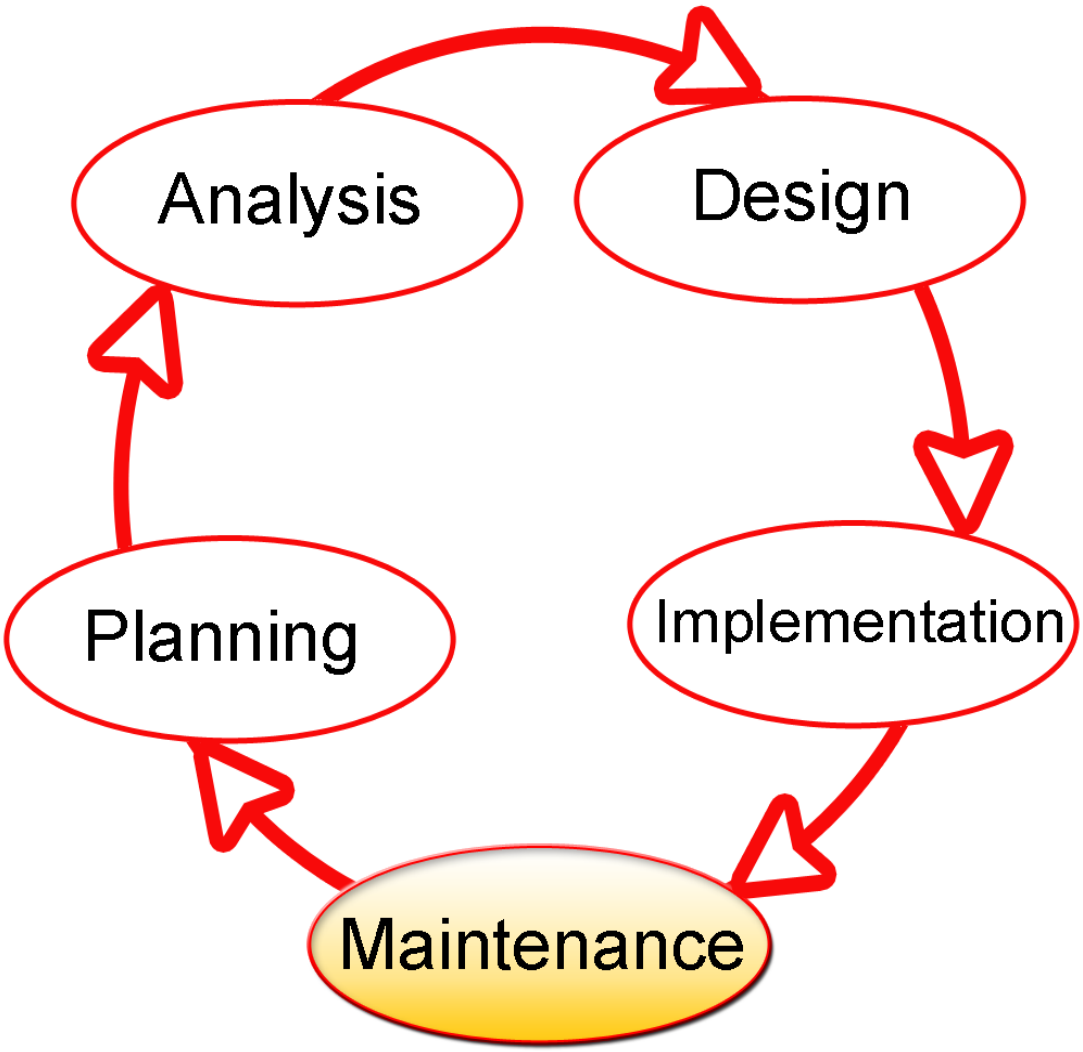
Overall, system design and development can be a powerful tool for addressing the needs of individuals with autism, by creating new technologies and support systems that can help them navigate the world around them and achieve their full potential. However, it is important to involve individuals with autism and their families in the design and development process, to ensure that the solutions created are tailored to their unique needs and preferences.

System Design

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It involves identifying the functionalities that the system must have, the resources required, and the constraints under which the system must operate. System design is critical to the development of software, hardware, and other systems that are capable of meeting the needs of users and stakeholders.

The process of system design typically involves several stages, including requirements analysis, system architecture design, module design, data design, and user interface design. Each of these stages involves a set of tasks, such as identifying user needs, defining system components and interfaces, designing data structures and algorithms, and creating user interfaces.

Good system design is essential for developing reliable, efficient, and scalable systems that can adapt to changing requirements and environments. It requires careful analysis, planning, and implementation to ensure that the system meets the needs of its users and stakeholders.


Analysis
Design
Implementation
Maintenance
Planning

a.INPUT DESIGN

Input design is the process of creating a plan or strategy for collecting data or information from various sources to support a specific business or research objective. The design of an effective input process requires careful consideration of the type of data needed, the sources of the data, the methods for collecting the data, and the tools and technology required to process and analyze the data.

The following are some key elements to consider when designing an effective input process:

1.Determine the objectives: Identify the specific information or data that is needed to achieve the business or research objective. This will help to determine the type of data required, the sources of data, and the methods for collecting the data.

2.Identify the sources: Determine the sources of data that are available, such as internal data sources (e.g., sales data, customer data, inventory data) and external data sources (e.g., market research, competitor data, industry reports).

3.Choose the method of data collection: Consider the most appropriate method for collecting the data, such as surveys, interviews, observation, or experiments. The method chosen should be efficient, accurate, and reliable.

4.Select the input device: Choose the input device that will be used to collect the data, such as a keyboard, mouse, touch screen, or scanner. The device should be appropriate for the type of data being collected and the method of data collection.

5.Design the input screen: Design the input screen or form that will be used to collect the data. The screen should be user-friendly, easy to navigate, and provide clear instructions to the user.

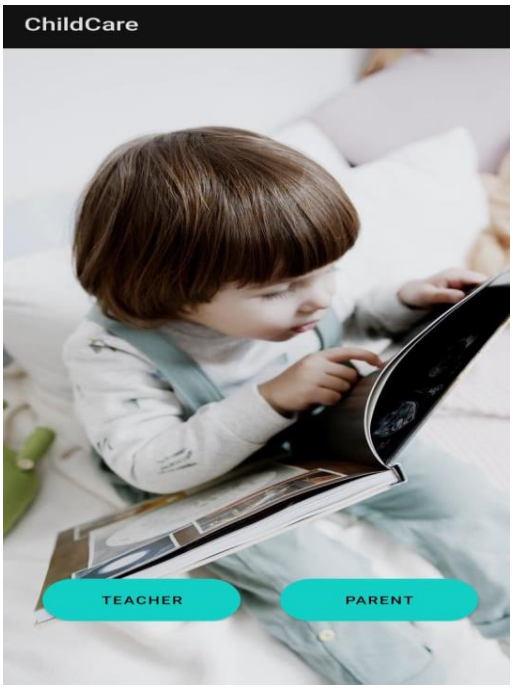
6.Validate the input: Implement validation rules to ensure that the data being entered is accurate and complete. This may include checks for data type, range, format, and consistency.

7.Test the input process: Test the input process to ensure that it is working correctly and efficiently. This may involve testing the data collection method, the input device, the input screen, and the validation rules.

By following these key elements, an effective input process can be designed that will provide accurate and reliable data to support business decisions or research objectives.

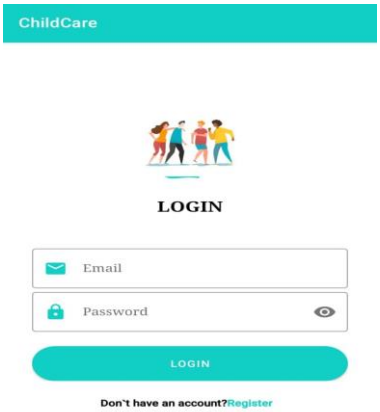
Form Design

Home Page

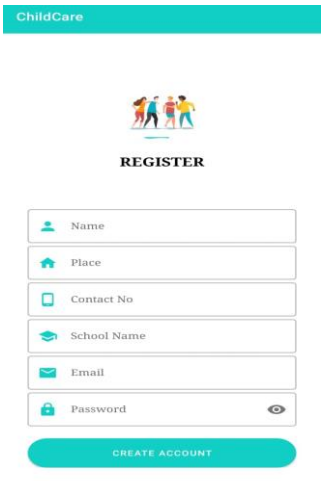


❖ TEACHER

Login Page

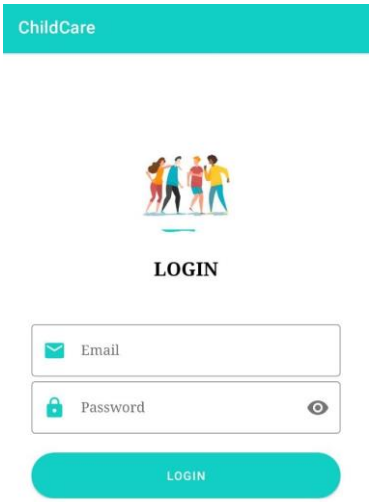


Registration Page

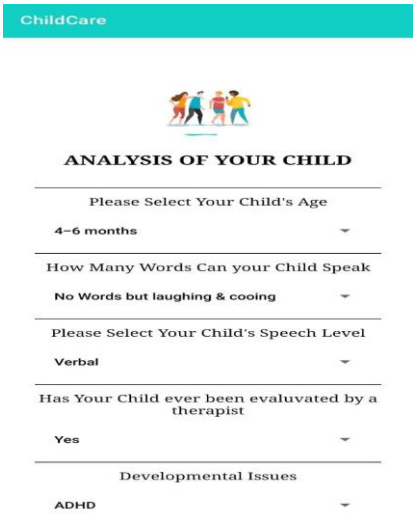


❖ PARENT

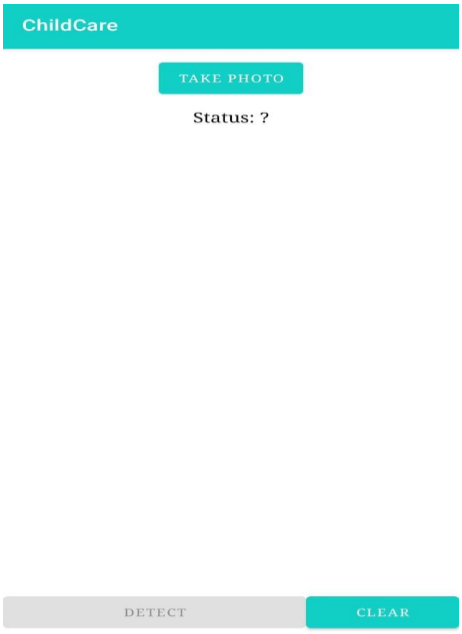
Login Page



Analysis of child



Face Recognition



b. OUTPUT DESIGN

The output design process involves identifying the output requirements of a system or a process, determining the format, content, and frequency of the output, and designing the output to meet those requirements.

Here are the steps involved in the output design process:

1.Define the output requirements: The first step is to define the output requirements of the system or process. This involves identifying the users of the output, the purpose of the output, and the frequency of the output.

2.Determine the format: Once the output requirements are defined, the next step is to determine the format of the output. The format should be selected based on the needs of the users, the type of information being presented, and the medium of delivery.

3.Define the content: The content of the output should be defined based on the requirements of the users. It should provide the necessary information in a clear and concise manner, and should be organized in a logical and meaningful way.

4.Design the layout: The layout of the output should be designed to make it easy for the users to read and understand. The layout should be consistent, and the use of fonts, colors, and graphics should be appropriate and consistent with the content.

5.Test and refine: Once the output design is complete, it should be tested to ensure that it meets the requirements of the users. Any necessary refinements should be made before the output is finalized.

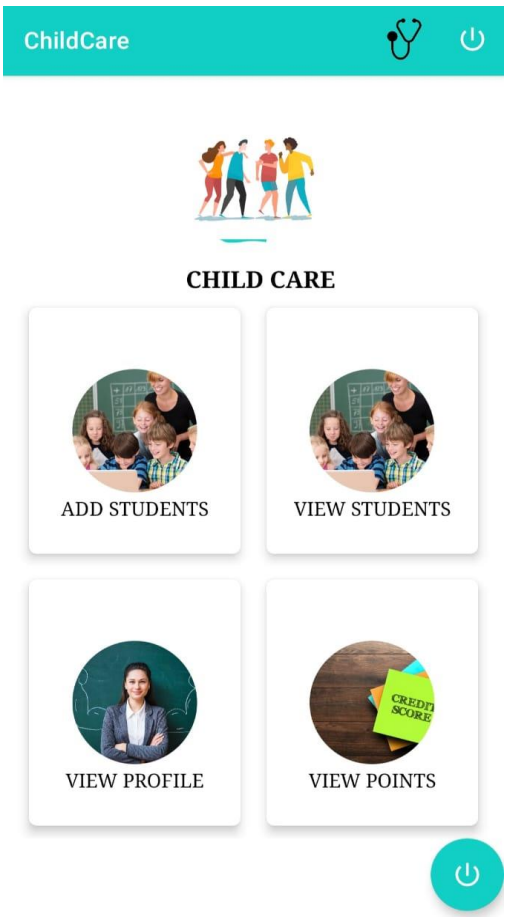
6.Implement: Once the output design is finalized, it can be implemented into the system or process. The users should be trained on how to use the output, and any necessary documentation should be provided to support its use.

Overall, the output design process involves a thorough understanding of the users' needs and requirements, and a careful consideration of the format, content, and layout of the output to ensure that it is effective in conveying the necessary information.

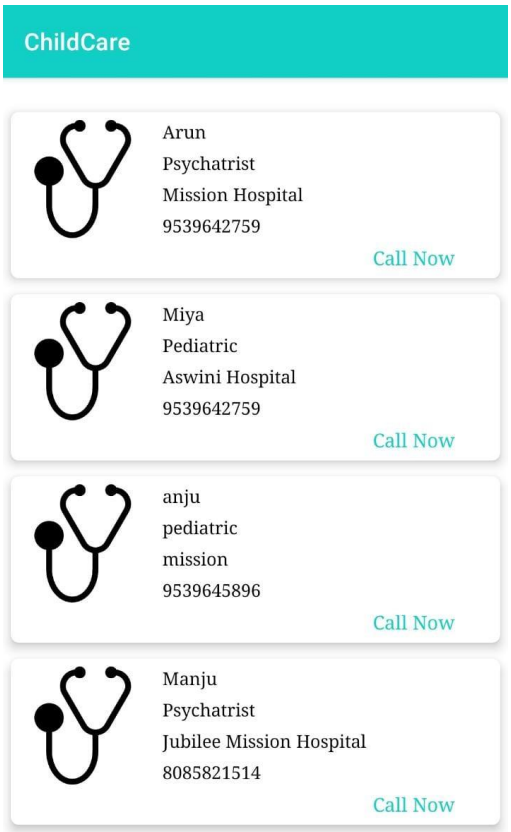
Form Design

❖ TEACHER

Student Details



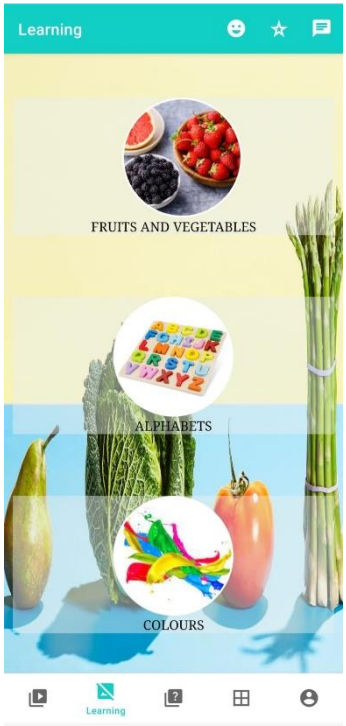
Doctor details



❖ PARENT

Entertainment

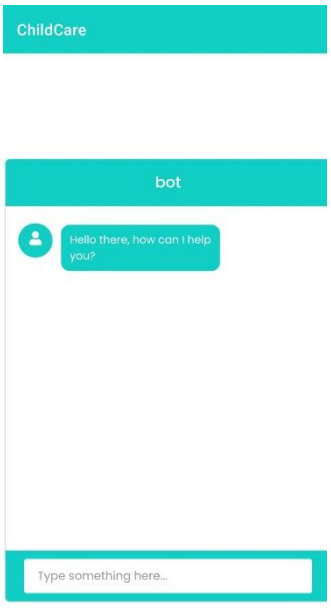
Learning



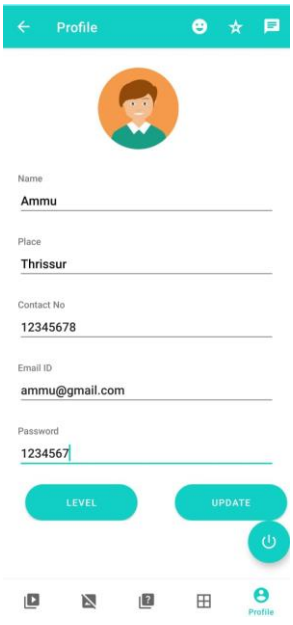
Quiz

Puzzle

Chat Bot



Profile



c. DATABASE DESIGN

Database design is the process of defining the structure, format, and relationships of data in a database. The design of a database involves determining what information will be stored in the database, how it will be organized, and how it will be accessed and manipulated.

Here are some key steps involved in designing a database:

1.Define the purpose of the database: Determine the goals and objectives of the database, what it is intended to achieve, and what types of data it will store.

2.Identify the entities and relationships: Identify the different entities or objects that will be stored in the database, and determine the relationships between them. This can be done using entity-relationship diagrams or other modeling techniques.

3.Determine the attributes: Determine the characteristics or attributes that describe each entity, such as name, address, and phone number. These attributes will become the fields or columns in the database.

4.Normalize the data: Normalize the data to eliminate redundancy and inconsistencies in the database. This involves breaking down the data into its smallest useful parts and organizing it into tables or entities.

5.Choose the database management system: Choose the appropriate database management system (DBMS) for your needs, such as MySQL, Oracle, or Microsoft SQL Server.

6.Define the data structures: Create the tables or entities in the database and define the data structures, such as data types, constraints, and indexes.

7.Populate the database: Populate the database with data, either manually or through automated means.

8.Test and refine: Test the database design and refine it as needed to ensure that it is efficient, accurate, and meets the needs of its users.

Effective database design is critical to ensuring that your data is well-organized, easily accessible, and accurate, and can be efficiently queried and analyzed to generate insights and support business decisions.

DFD (Data Flow Diagram)

Data Flow Diagram is a graphical representation of the system specification. It is a network that uses Special symbols to describe the flow of data and process that transforms data throughout a system. Four basic symbols are used to construct data flow diagrams; these symbols represent data sources, data flows, data transformations, and data storage.

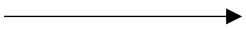
The design has been described as a multi-step process in which representations of data structure, program structure, interface characteristics, and procedural details are synthesized from information requirements. The design phase begins when the Requirements Specification Document for the software to be developed, is available. System design is a creative and challenging phase that serves as a foundation for all software engineering and maintenance steps that follows. The design builds coherent, well-planned representations of programs. The design specification describes the features of the system, the components or elements of the system and their appearance to users. In the design phase, data flow, data-stores, processes, procedures, controls etc. are designed.

Data Flow Diagram represents one of the most ingenious tools used for structured analysis. A Data flow diagram or DFD as it is shortly called is also known as a bubble chart. It has the purpose of clarifying system requirements and identifying major transformation, which will become a program in system design. It is the major starting point in the design phase that functionally decomposes the requirements specification down to the last level of detail.

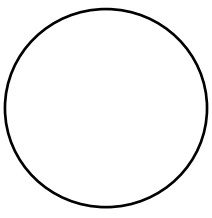
The basic elements of DFD are: -

- Bubble: Used to represent a function.
- Arrows: Used to represent data flow.
- Rectangle: Used to represent external entities.
- Open rectangle: Used to represent data store.

Basic Data Flow Diagram Notations



A data flow is a route, which enables packets of data to travel from one point to another. Data may flow from a source to a processor and from data store or process. An arrow line depicts the flow, withanarrowhead pointing in the direction of flow.



A process represents transformation where incoming data flows are changed into outgoing data flows.

A data store is a repository of data that is to be stored for use by one or more process may be as simple as a buffer or queue or sophisticated as a relational database.

A source or sink is a person or part of an organization, which enters or receives information from the system, but is outside the contest of data flow model.

Steps to construct Data Flow Diagram:

Four steps are basically used to construct DFD,

- Process should be named a number for easy reference. Each name should be representative of the process.
- The direction of flow is from top to bottom and from left to right.
- When a process is exploded into lower level details, they are numbers.
- The name of data stores, sources and destination are written in capital letters.

DFDs are divided into 2 levels. They are as follows:

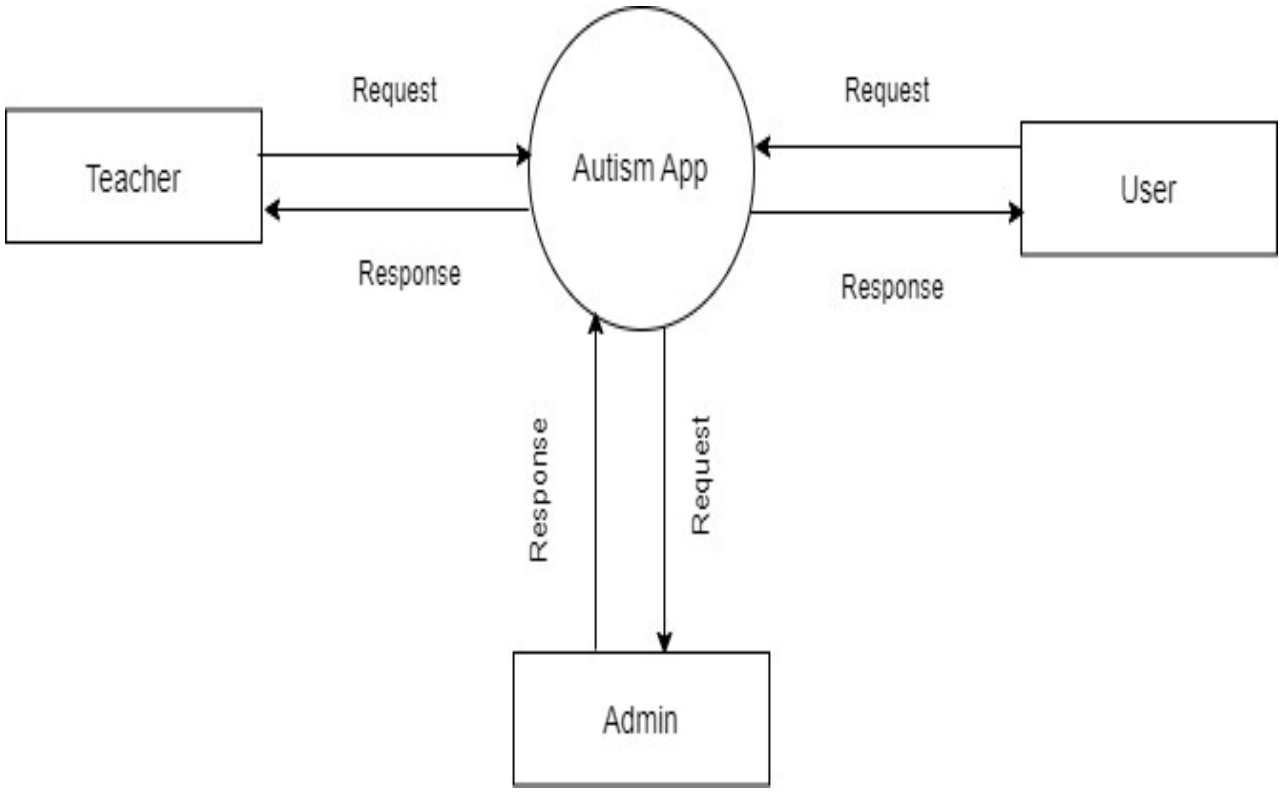
1. Level 0

It gives us the description about the users that login in the software. It shows the initial level of the state of data and depicts the interaction of the software and with its corresponding database. The database is the collection of tables, procedures, views and all the required data for the same.

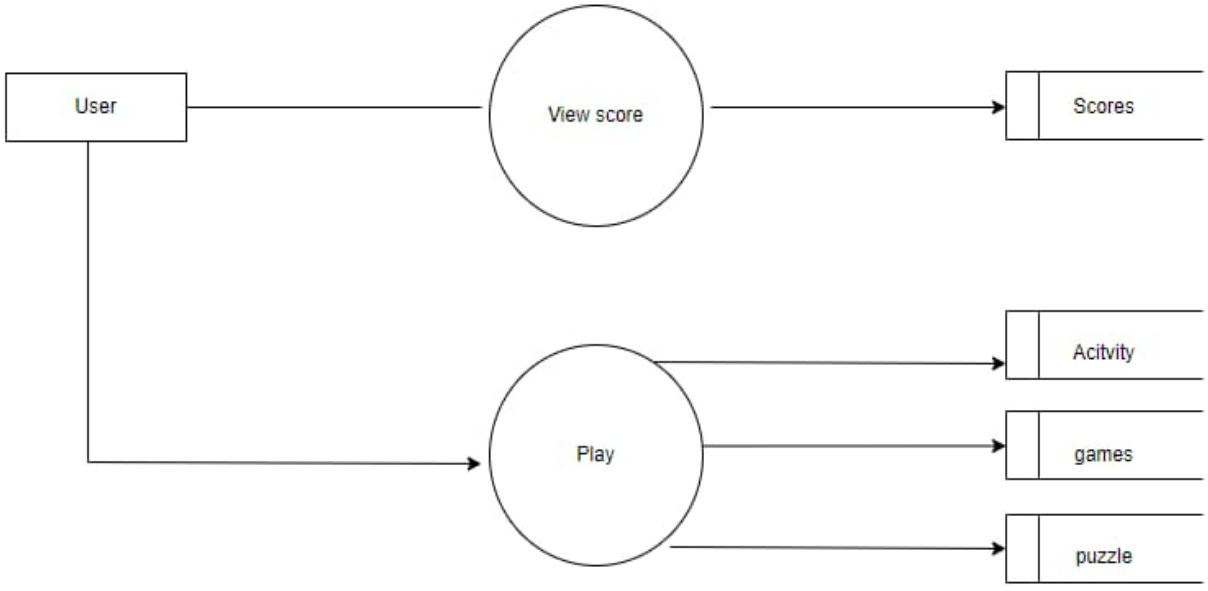
2. Level 1

The level 1 DFD shows how the system is divided into subsystems, each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole.

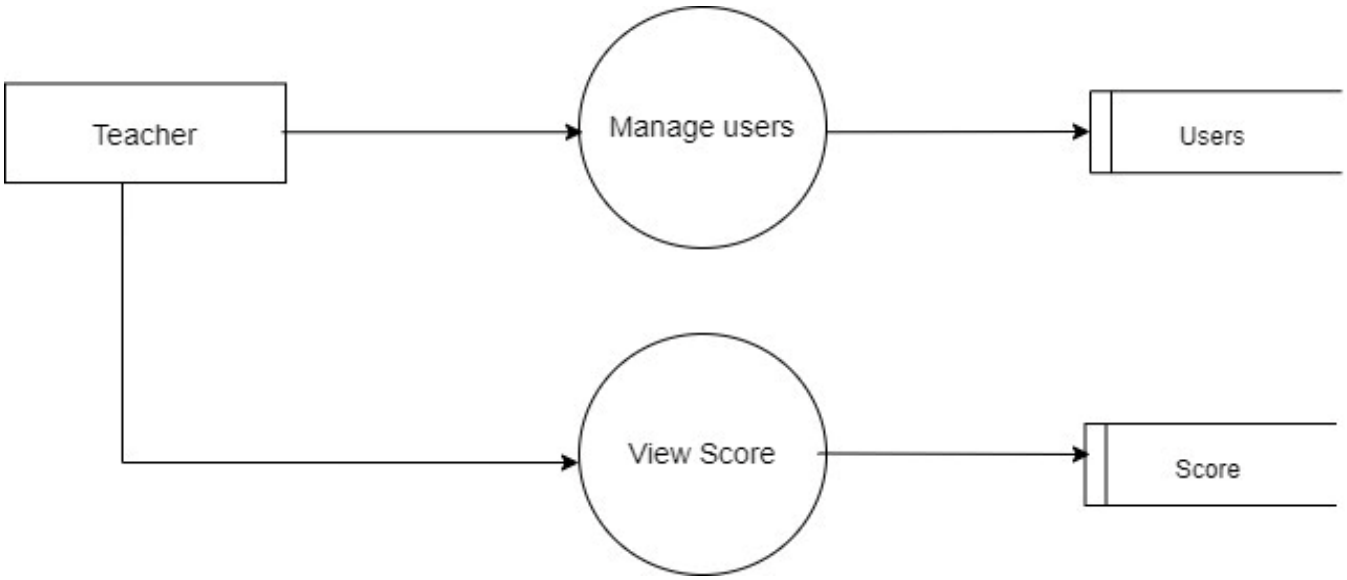
DFD LEVEL-0


Teacher
Request
Response
Autism App
Request
Response
User
Response
Request
Admin

LEVEL 1 USER



LEVEL 1 TEACHER



ER DIAGRAM

An ER diagram, or Entity-Relationship diagram, is a type of visual representation used to model the relationships between entities in a database. The purpose of an ER diagram is to provide a high-level overview of the database schema, showing the relationships between tables, their attributes and the cardinality of these relationships. ER diagrams are widely used in database design and development, and are a key component of the data modeling process.

ER diagrams consist of three main components:

1.Entities: These are the objects or concepts represented in the database. Examples might include customers, orders, products, and suppliers.

2.Relationships: These represent the connections or associations between entities. Relationships can be one-to-one, one-to-many, or many-to-many.

3.Attributes: These are the characteristics or properties of an entity, such as name, address, and phone number.

In an ER diagram, entities are represented as rectangles, relationships as diamonds, and attributes as ovals or circles. The lines connecting these shapes indicate the type of relationship between them, and may include labels to indicate the cardinality of the relationship.

Overall, an ER diagram provides a clear and concise way to represent the structure of a database, making it easier to understand and manage.

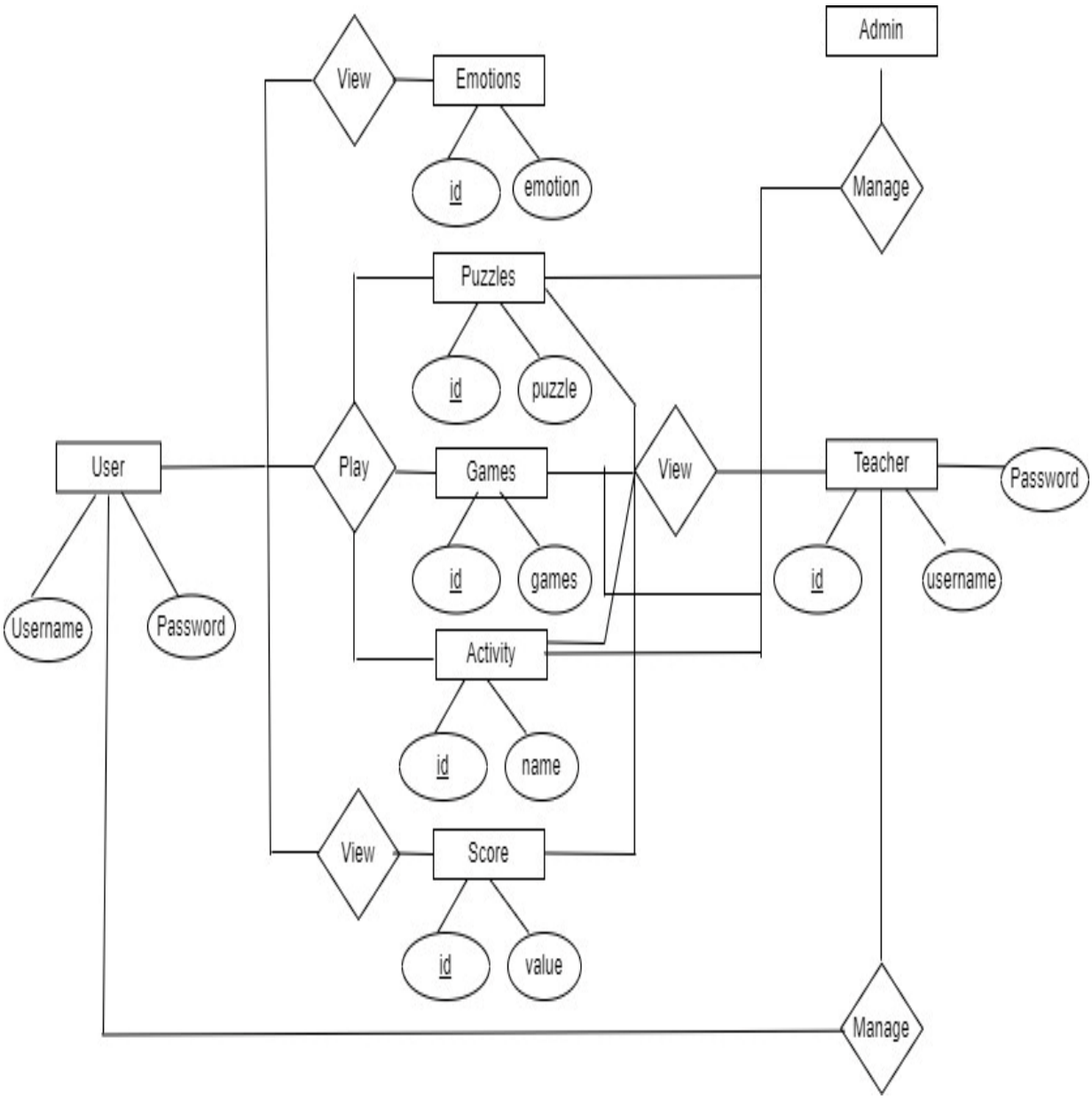

Admin
View
Emotions
id
emotion
Manage
Puzzles
id
puzzle
User
Play
Games
View
Teacher
Password
id
games
id
username
Username
Password
Activity
id
name
View
Score
id
value
Manage

TABLE DESIGN

A table is a collection of related data held in a structures format within a database. It consists of columns and rows. In a relational database, a table is set of data elements using a model of vertical columns and horizontal rows, the cell being the unit where a row and column intersect. A table has a specified number of columns, but can have any number of rows. Each row is identified by one or more values appearing in a particular column subset. The columns subset, which uniquely identifies a row, is called the primary key.

"Table" is another term for "relation"; although there is the difference in that a table is usually a multi set of rows where as a relation is a set and does not allow duplicates. Besides the actual data rows, tables generally have associated with them some methods such as constraints on the table or on the values within particular columns.

❖ Admin

#	Name	Type	Collation	Attributes	Null	Default
1	**Id(primarykey)**	Int(11)			No	None
2	**Username**	Varchar(200)		utf_8unicode_ci	No	None
3	**Password**	Varchar(200)		utf_8unicode_ci	No	None

❖ Alphabets

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**cid(primarykey)**	Int(11**)**			No	None		AUTO_INCREMENT
2	**image**	Varchar(1000)	utf_8unicode_ci		No	None		
3	**text**	Varchar(200)	utf_8unicode_ci		No	None		
4	**points**	Varchar(200)	utf_8unicode_ci		No	None		

❖ Alphabet Puzzle

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**aid(primarykey)**	int(11)			No	Null		AUTO_INCREMENT
2	**question**	Varchar(300)	utf_8unicode_ci		No	Null		
3	**Option1**	Varchar(200)	utf_8unicode_ci		No	Null		
4	**Option2**	Varchar(200)	utf_8unicode_ci		No	Null		
5	**Option3**	Varchar(200)	utf_8unicode_ci		No	Null		
6	**answer**	Varchar(200)	utf_8unicode_ci		No	Null		

❖ Colours

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**cid(primarykey)**	Int(11)			NO	None		AUTO_INCREMENT
2	**image**	Varchar(1000)	utf_8unicode_ci		NO	None		
3	**text**	Varchar(200)	utf_8unicode_ci		NO	None		
4	**points**	Varchar(200)	utf_8unicode_ci		NO	None		

❖ Doctors

#	Name	Type	Collation	Attributes	Null	Default
1	**did(primarykey)**	int(11)			NO	None
2	**name**	Varchar(200)	utf_8unicode_ci		NO	None
3	**special**	Varchar(200)	utf_8unicode_ci		NO	None
4	**hospital**	Varchar(200)	utf_8unicode_ci		NO	None
5	**contact**	Varchar(200)	utf_8unicode_ci		NO	None
6	**gender**	Varchar(100)	utf_8unicode_ci		NO	None
7	**workingtime**	Varchar(100)	utf_8unicode_ci		NO	None
8	**fee**	Varchar(100)	utf_8unicode_ci		NO	None
9	**password**	Varchar(100)	utf_8unicode_ci		NO	None

❖ Music

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**mid(primarykey)**	int(11)			NO	None		AUTO_INCREMENT
2	**music**	Varchar(6000)	utf_8unicode_ci		NO	None		

❖ Numbers

#	Name	Type	Collation	Attributes	Null	Default
1	**nid(primarykey)**	int(11)			NO	None
2	**numbers**	Varchar(200)	utf_8unicode_ci		NO	None
3	**Option1**	Varchar(200)	utf_8unicode_ci		NO	None
4	**Option2**	Varchar(200)	utf_8unicode_ci		NO	None
5	**Option3**	Varchar(200)	utf_8unicode_ci		NO	None
6	**answers**	Varchar(200)	utf_8unicode_ci		NO	None

❖ Points

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**pid(primarykey)**	int(11)			NO	None		AUTO_INCREMENT
2	**userid**	Varchar(200)	utf_8unicode_ci		NO	None		
3	**points**	Varchar(200)	utf_8unicode_ci		NO	None		

❖ Puzzle

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**pid(primarykey)**	int(11)			NO	None		AUTO_INCREMENT
2	**Image1**	Varchar(1000)	utf_8unicode_ci		NO	None		
3	**Image2**	Varchar(1000)	utf_8unicode_ci		NO	None		
4	**Image3**	Varchar(1000)	utf_8unicode_ci		NO	None		
5	**Image4**	Varchar(1000)	utf_8unicode_ci		NO	None		

❖ Recognize

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**cid(primarykey)**	int(11)			NO	None		AUTO_INCREMENT
2	**image**	Varchar(1000)	utf_8unicode_ci		NO	None		
3	**text**	Varchar(300)	utf_8unicode_ci		NO	None		
4	**points**	Varchar(300)	utf_8unicode_ci		NO	None		

❖ QUIZ

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**qid(primarykey)**	int(11)			NO	None		AUTO_INCREMENT
2	**question1**	Varchar(200)	utf_8unicode_ci		NO	None		
3	**option1q1**	Varchar(200)	utf_8unicode_ci		NO	None		
4	**option2q1**	Varchar(200)	utf_8unicode_ci		NO	None		
5	**option3q1**	Varchar(200)	utf_8unicode_ci		NO	None		
6	**answerq1**	Varchar(200)	utf_8unicode_ci		NO	None		
7	**question2**	Varchar(200)	utf_8unicode_ci		NO	None		
8	**Option1q2**	Varchar(200)	utf_8unicode_ci		NO	None		
9	**Option2q2**	Varchar(200)	utf_8unicode_ci		NO	None		
10	**Option3q2**	Varchar(200)	utf_8unicode_ci		NO	None		
11	**answerq2**	Varchar(200)	utf_8unicode_ci		NO	None		
12	**question3**	Varchar(200)	utf_8unicode_ci		NO	None		
13	**option1q3**	Varchar(200)	utf_8unicode_ci		NO	None		
14	**option2q3**	Varchar(200)	utf_8unicode_ci		NO	None		
15	**option3q3**	Varchar(200)	utf_8unicode_ci		NO	None		
16	**answerq3**	Varchar(200)	utf_8unicode_ci		NO	None		

❖ Register

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id(primarykey)**	int(11)			NO	None		AUTO_INCREMENT
2	**name**	Varchar(200)	utf_8unicode_ci		NO	None		
3	**mobile**	Varchar(200)	utf_8unicode_ci		NO	None		
4	**place**	Varchar(200)	utf_8unicode_ci		NO	None		
5	**email**	Varchar(200)	utf_8unicode_ci		NO	None		
6	**password**	Varchar(200)	utf_8unicode_ci		NO	None		
7	**status**	Varchar(200)	utf_8unicode_ci		NO	None		
8	**teacherid**	Varchar(200)	utf_8unicode_ci		NO	None		

❖ Teacher_Register

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**tid(primarykey)**	int(11)			NO	None		AUTO_INCREMENT
2	**name**	Varchar(200)	utf_8unicode_ci		NO	None		
3	**mobile**	Varchar(200)	utf_8unicode_ci		NO	None		
4	**place**	Varchar(200)	utf_8unicode_ci		NO	None		
5	**email**	Varchar(200)	utf_8unicode_ci		NO	None		
6	**password**	Varchar(200)	utf_8unicode_ci		NO	None		
7	**school**	Varchar(200)	utf_8unicode_ci		NO	None		
8	**status**	Varchar(200)	utf_8unicode_ci		NO	None		

❖ Videos

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**vid(primarykey)**	int(11)			NO	None		AUTO_INCREMENT
2	**title**	Varchar(200)	utf_8unicode_ci		NO	None		
3	**description**	Varchar(200)	utf_8unicode_ci		NO	None		
4	**video**	Varchar(200)	utf_8unicode_ci		NO	None		

❖ Verify

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**vid(primarykey)**	int(11)			NO	None		AUTO_INCREMENT
2	**age**	Varchar(200)	utf_8unicode_ci		NO	None		
3	**words**	Varchar(200)	utf_8unicode_ci		NO	None		
4	**speech**	Varchar(200)	utf_8unicode_ci		NO	None		
5	**therapy**	Varchar(200)	utf_8unicode_ci		NO	None		
6	**development**	Varchar(200)	utf_8unicode_ci		NO	None		
7	**toys**	Varchar(200)	utf_8unicode_ci		NO	None		
8	**puzzle**	Varchar(200)	utf_8unicode_ci		NO	None		

9	**circle**	Varchar(200)	utf_8unicode_ci		NO	None		
10	**tower**	Varchar(200)	utf_8unicode_ci		NO	None		
11	**body**	Varchar(200)	utf_8unicode_ci		NO	None		
12	**command**	Varchar(200)	utf_8unicode_ci		NO	None		
13	**picture**	Varchar(200)	utf_8unicode_ci		NO	None		

❖ Chatbot

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id(primarykey)**	int(11)			NO	None		AUTO_INCREMENT
2	**queries**	Varchar(300)	latin_swedish_ci		NO	None		
3	**replies**	Varchar(300)	latin_swedish_ci		NO	None		

SYSTEM TESTING & IMPLEMENTATION

SYSTEM TESTING

user acceptance. In other word software testing is a System testing is the evaluation of software item to detect difference between given input and expected output. Also, to assess the feature of software item. Testing assesses the quality of the product. Software Testing is the process that should be done during the development process. It offers the greatest security since the old system can take over the errors is found or inability to handle certain type of transactions while using the new system. So testing is vital to the success of new system. Series of the system are performed for the proposed system before the system is ready for verification and validation process.

TESTING METHODOLOGIES

These exists different methodologies in testing to make sure phase. And to check whether it satisfies the specified requirements at the end of the development phase.

The major testing strategies are:

1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

❖ Unit Testing

Unit Testing is a level of software testing where individual units/ components of the software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of the software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class, or derived/ child class. (Some treat a module of an

application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Once all the units in a program have been found to be working in the most efficient and error-free manner possible, larger components of the program can be evaluated by means of integration testing.

❖ Integration Testing

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and interposes communication is tested and individual subsystems are exercised through their input interface.

Some different types of integration testing are a big bang, top-down, and bottom-up, mixed (sandwich) and risky hardest. Other Integration Patterns [2] are collaboration integration, backbone integration, and layer. Integration, client-server integration, distributed services integration, and high-frequency integration.

Upon completion of unit testing, the units or modules are to be integrated which gives rise to integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

❖ System Testing

- System testing is performed based on written test cases according to information collected from detailed architecture/design document, module specification, and requirement specification.
- System testing may be started one unit, component, and integration testing are completed.

❖ WHITE BOX TESTING

White Box Testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code Based Testing or Structural Testing) is a software testing method in which the internal structure/ design/ implementation of the item being tested is

known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming knows how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system. This method is named so because the software program, in the eyes of the tester, is like a white/ transparent box; inside which one clearly sees.

White-box testing: Testing based on an analysis of the internal structure of the component or system.

White-box test design technique: Procedure to derive and/or select test cases based on an analysis of the internal structure of a component or system.

White box testing advantages:

- Testing can be commenced at an earlier stage. One need not wait for the GUI to be available.
- Testing is more thorough, with the possibility of covering most paths.

White Box Testing Disadvantages*:*

- Since tests can be overly complex, highly skilled resources are required, with a thorough knowledge of programming and implementation.
- Test script maintenance can be a burden if the implementation changes too frequently.
- Since this method of testing is closely tied with the application being testing, tools to cater to every kind of implementation/platform may not be readily available.

❖ BLACK BOX TESTING

Black Box Testing, also known as Behavioral Testing, is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. These tests can be functional or nonfunctional, though usually functional. This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behavior or performance errors

- Initialization and termination errors

Black Box Testing Advantages

- Tests are done from a user's point of view and will help in exposing discrepancies in the specifications.
- Tester need not know programming languages or how the software has been implemented.
- Tests can be conducted by a body independent from the developers, allowing for an objective perspective and the avoidance of developer bias.
- Test cases can be designed as soon as the specifications are complete.

Black Box Testing Disadvantages

- Only a small number of possible inputs can be tested and many program paths will be left untested.
- Without clear specifications, which are the situation in many projects, test cases will be difficult to design.
- Tests can be redundant if the software designer/ developer has already run a test case.

❖ Acceptance Testing

Acceptance testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. Formal testing with respect to user needs, requirements, and business process conducted to determine whether a system satisfies the acceptance criteria and to enable the user, customer or other authorized entity to determine whether or not to accept the system.

- Usually, black box testing method is used in acceptance testing. Testing does not normally follow a strict procedure and is not scripted but is rather ad-hoc. Acceptance testing is a testing technique performed to determine whether the software system has met the requirements specifications

IMPLEMENTATION AND TESTING

❖ ANDROID SDK

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in Java programming language using the Android Software Development Kit (SDK), but other development environments are also available. The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS 10.5.8 or later, and Windows XP or later. The SDK is not currently available on Android; however, the software can be developed by using specialized Android application

Until around the end of 2014, the officially supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. As of 2015, Android Studio, made by Google and powered by IntelliJ, is the official IDE; however, developers are free to use others. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and

tools can also be downloaded for compatibility testing. Android applications are packaged in apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains dex files (compiled bytecode files called Dalvik executable), resource files, etc.

FRONT END TOOLS: ANDROID – JAVA

❖ ANDROID

Android is a mobile operating system developed by Google. It is based on a modified version of the Linux kernel and other open-source software, and is designed primarily for touchscreen mobile devices such as smartphones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics. One of the most widely used mobile OS these days is ANDROID. Android is a software bunch comprising not only operating system but also middleware and key applications. Google's Android division certainly has a sense of humor: It named all of its version codenames after desserts

FEATURES OF ANDROID

Near Field Communication (NFC):Most Android devices support NFC, which allows electronic devices to easily interact across short distances. The main aim here is to create a payment option that is simpler than carrying credit cards or cash, and while the market hasn't exploded as many experts had predicted, there may be an alternative in the works, in the form of Bluetooth Low Energy (BLE).

Infrared Transmission**:** The Android operating system supports a built-in infrared transmitter, allowing you to use your phone or tablet as a remote control.

Automation**:** The Tasker app lets you not only control app permissions but also automate them.

Alternate Keyboards**:** Android supports multiple keyboards and makes them easy to install; the SwiftKey, Skype, and 8pen apps all offer ways to quickly change up your keyboard style. Other mobile operating systems either don't permit extra keyboards at all, or the process to install and use them are tedious and time-consuming.

Storage and Battery Swap**:** Android phones also have unique hardware capabilities. Google's OS makes it possible to remove and upgrade your battery or to replace one that no longer holds a charge. In addition, Android phones come with SD card slots for expandable storage.

Custom Home Screens**:** While it's possible to hack certain phones to customize the home screen, Android comes with this capability from the get-go. Download a third-party launcher like Nova, Apex or Slide and you can add gestures, new shortcuts, or even performance enhancements for older-model devices.

❖ JAVA

Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platform specific machines while Java is a write once, run anywhere

language. A platform is the hardware or software environment in which a program runs. Java is one of the world's most important and widely used computer languages, and it has held this distinction for many years. Object Oriented meaning the capability to reuse code. It is possible to develop a single application which can run on multiple platforms like Windows, UNIX, and Macintosh systems. Java does not support the use of pointers. It automatically manages memory garbage-collection routine is activated when system runs short of memory. Java provides the most secure programming environment. Java doesn't just fix security loopholes-it eliminates them, which makes Java the perfect language for programming on the Web.

FEATURES OF JAVA

- Simple **:**Java is an extension of C and C++ with the added feature of garbage collection and improved memory management.
- Object oriented**:** Object-oriented programming deals with objects and their behaviors and hence an analogy of the real world can be found in programs.
- Network**:** Java has an extensive library of routines for coping easily with TCP/IP protocols like HTTP and FTP. This makes creating network connections much easier.
- Robust**:** Java is intended for writing programs that must be reliable in a variety of ways. Java puts a lot of emphasis on early checking for possible problems, later dynamic (runtime) checking, and eliminating situations that are error-prone.
- Secure: Java is intended for use in networked/distributed environments. Toward that end, a lot of emphases have been placed on security. The changes to the semantics of pointers make it impossible for applications to forge access to the user's hard disk.
- Portable**:** There are no "implementation dependent" (machine/ processor dependent) aspects of the specification. The sizes of the primitive data types (integer, float) are specified.

BACK-END TOOL: MYSQL – PHP

❖ MYSQL

MySQL is an open source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation

for Structured Query Language. MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing. MySQL is an important component of an open source enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web server, MySQL as the relational database management system and PHP as the object-oriented scripting language.

- Originally conceived by the Swedish company MySQL AB, MySQL was acquired by Sun Microsystems in 2008 and then by Oracle when it bought Sun in 2010.

Developers can use MySQL under the GNU General Public License (GPL), but enterprises must obtain a commercial license from Oracle.

Today, MySQL is the RDBMS behind many of the top websites in the world and countless corporate and consumer-facing web-based applications, including Facebook, Twitter and YouTube.

FEATURES OF MYSQL

- **Relational Database Management System (RDBMS):** MySQL is a relational database management system.
- **Easy to use:** MySQL is easy to use. You have to get only the basic knowledge of SQL. You can build and interact with MySQL with only a few simple SQL statements.
- **It is secure:** MySQL consist of a solid data security layer that protects sensitive data from intruders. Passwords are encrypted in MySQL.
- **Client/ Server Architecture:** MySQL follows a client /server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc.

- **Free to download:** MySQL is free to use and you can download it from MySQL official website.
- **It is scalable:** MySQL can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, you can increase this number to a theoretical limit of 8 TB of data.

❖ PHP

PHP was at first created as a simple scripting platform called "Personal Home Page". Nowadays PHP is an alternative of the Microsoft's Active Server Pages (ASP) technology.PHP is an open source server-side language which is used for creating dynamic web pages. It can be embedded into HTML. PHP is usually used in conjunction with a MySQL database on Linux/UNIX web servers. It is probably the most popular scripting language.

PHP is a widely-used general-purpose scripting language and interpreter that is freely available. A full explanation of all the PHP functions, complete user manual and lots of tutorials can be found on the PHP's official page.PHP code may be executed with a command line interface (CLI), embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks.

FEATURES OF PHP

- **Interpreted :**It is an interpreted language, i.e. there is no need for compilation.
- **Faster :**It is faster than other scripting language e.g. asp and jsp.
- **Open Source :**Open source means you no need to pay for use php, you can free download and use.

- **Platform Independent :**PHP code will be run on every platform, Linux, Unix, Mac OS X, Windows.
- **Case Sensitive :**PHP is case sensitive scripting language at time of variable declaration. In PHP, all keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are NOT case-sensitive.
- **Error Reporting :**PHP have some predefined error reporting constants to generate a warning or error notice.
- **Real-Time Access Monitoring :**PHP provides access logging by creating the summary of recent accesses for the user.

OVERVIEW OF PACKAGES

❖ MySQL Database

MySQL is a fast easy- to- use RDBMS being used for many small and big businesses. MySQL is developed, supported, and marketed by MySQL LAB, which is a Swedish company.MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.

- MySQL is released under an open –source license. So,you have nothing to pay for use it.
- MySQL is very powerful program in its own right. It handles a large subset of functionality of the most expensive and powerful database packages.
- MySQL work very quickly and works well even with large data sets.
- MySQL works on any operating systems and within many languages including JAVA, PHP, C, C++, Pearl etc.
- MySQL uses the standard form of the well-known SQL data language.

SCOPE

SCOPE

ASD is a spectrum disorder, meaning that it affects people differently and to varying degrees. Some individuals with ASD may have mild symptoms and require little support, while others may have severe symptoms and require significant support.

The symptoms of ASD typically appear in early childhood, and diagnosis is usually made by the age of 2 or 3. Some common symptoms of ASD include:

- Difficulty with social interactions, such as difficulty making eye contact, understanding social cues, or engaging in back-and-forth conversation
- Restricted interests and repetitive behaviors, such as lining up objects, repeating words or phrases, or becoming fixated on certain objects or topics
- Sensory sensitivities, such as being bothered by certain sounds, textures, or tastes
- Difficulty with communication, such as delayed speech or difficulty expressing oneself

ASD is a lifelong disorder, but with appropriate support and interventions, individuals with ASD can lead fulfilling and productive lives. Treatment for ASD typically involves a combination of behavioral therapy, medications, and educational interventions, tailored to the individual's needs and strengths.

An ASD app can have a broad scope, as there are many different aspects of autism spectrum disorder (ASD) that can be addressed through digital technology. Some potential areas of focus for an ASD app include:

1.Communication: Many individuals with ASD struggle with communication, so an app that helps them communicate more effectively could be very helpful. This could include features such as visual schedules, picture exchange communication systems (PECS), and text-to-speech functionality.

2.Social skills: Another common difficulty for individuals with ASD is social skills. An app could provide social stories, role-playing scenarios, or other interactive activities to help individuals with ASD practice social skills and improve their ability to interact with others.

3.Sensory sensitivities: Some individuals with ASD are hypersensitive to certain stimuli, such as loud noises or bright lights. An app could provide tools for managing sensory overload, such as white noise generators or sensory input regulators.

4.Self-regulation: Many individuals with ASD struggle with self-regulation, meaning they have difficulty controlling their emotions and behavior. An app could provide tools for self-monitoring, self-calming, and self-reflection to help individuals with ASD manage their emotions and behavior.

5.Education: An app could provide educational resources for individuals with ASD, such as videos, articles, and interactive activities that explain the symptoms and management of ASD. The scope of an ASD app will depend on the specific goals and features of the app, as well as the needs of the individuals with ASD who will be using it. However, a well-designed ASD app can provide valuable support for individuals with ASD and their families.

a. FUTURE ENHANCEMENTS

ASD, or autism spectrum disorder, is a complex neurodevelopmental disorder that affects social communication, behavior, and cognitive development. There are many ongoing efforts to enhance our understanding of ASD and to develop better interventions and treatments for individuals with ASD. Here are some possible future enhancements for ASD:

1.Early detection and intervention: Early detection and intervention can greatly improve outcomes for individuals with ASD. Advances in screening tools and techniques may help to identify children at risk of developing ASD at an earlier age, allowing for earlier intervention and treatment.

2.Personalized treatment plans: Every individual with ASD is unique and requires a personalized treatment plan that addresses their specific needs. Advances in technology, such as machine learning algorithms, may help to develop personalized treatment plans based on individual characteristics and behaviors.

3.Integration of behavioral and pharmacological interventions: Behavioral interventions, such as Applied Behavior Analysis (ABA), are effective in improving social and communication skills in individuals with ASD. However, pharmacological interventions may also be beneficial in managing co-occurring conditions such as anxiety and depression. Integrating both approaches may lead to better outcomes for individuals with ASD.

4.Improved access to services: Access to services, such as therapy and education, can be a challenge for individuals with ASD and their families. Improvements in telehealth and teletherapy may increase access to services for individuals in rural or underserved areas.

5.Increased understanding of the genetics of ASD: Advances in genetics research may help to identify the specific genetic mutations that contribute to the development of ASD. This knowledge may lead to more targeted treatments and therapies for individuals with ASD.

6.Greater understanding of the neural mechanisms of ASD: Advances in neuroimaging techniques may help to identify the neural mechanisms underlying ASD. This knowledge may lead to the development of new treatments and therapies that target specific brain regions or circuits.

7.Development of new therapies: There is currently no cure for ASD, but there are many ongoing efforts to develop new therapies and treatments. For example, stem cell therapy and gene therapy are being investigated as potential treatments for ASD.

Overall, the future enhancements of ASD will likely involve a combination of advances in technology, increased understanding of the underlying biology of ASD, and a greater focus on personalized treatment plans that address the unique needs of each individual with ASD.

CONCLUSION

CONCLUSION

ASD (autism spectrum disorder) is a complex neurodevelopmental condition that affects social interaction, communication, and behavior. It is a spectrum disorder, which means that the severity of symptoms can vary widely between individuals, from mild to severe.

There is no known cure for ASD, but early intervention and therapy can greatly improve outcomes for individuals with the condition. Treatment options include behavioral therapy, speech therapy, occupational therapy, and medication to manage co-occurring conditions.

Research into the causes of ASD is ongoing, and it is believed that a combination of genetic and environmental factors play a role in its development. However, much more research is needed to fully understand the underlying causes of ASD.

It is important to recognize and support individuals with ASD, as they can face significant challenges in navigating social situations and daily life. With the right support and understanding, individuals with ASD can lead fulfilling and successful lives.

ASD (autism spectrum disorder) apps can be helpful for individuals with autism, as they can provide support for various aspects of life, including communication, social skills, and sensory processing. It is important to note that each individual with ASD is unique and may require different types of support, and therefore, it is crucial to work with a healthcare professional to determine the best course of treatment and support. Overall, technology can be a valuable tool for individuals with ASD, but it should be used in conjunction with other therapies and interventions.

BIBLIOGRAPHY

BIBLIOGRAPHY

- American Psychiatric Association. (2013). Diagnostic and statistical manual of mental disorders (5th ed.). Arlington, VA: American Psychiatric Publishing.
- Autism Speaks. (2021). What is Autism? Retrieved from https://www.autismspeaks.org/what-autism
- Centers for Disease Control and Prevention. (2021). Data and statistics on autism spectrum disorder. Retrieved from https://www.cdc.gov/ncbddd/autism/data.html
- Dawson, G., & Burner, K. (2011). Behavioral interventions in children and adolescents with autism spectrum disorder: A review of recent findings. Current Opinion in Pediatrics, 23(6), 616–620.
- Lai, M. C., Lombardo, M. V., & Baron-Cohen, S. (2014). Autism. Lancet, 383(9920), 896–910.
- National Institute of Mental Health. (2021). Autism spectrum disorder. Retrieved from https://www.nimh.nih.gov/health/topics/autism-spectrum-disorders-asd/index.shtml
- Ozonoff, S., & Dawson, G. (2010). Early identification of autism spectrum disorder: Recommendations for practice and research. Pediatrics, 125(1), 118–127.
- Rogers, S. J., & Vismara, L. A. (2008). Evidence-based comprehensive treatments for early autism. Journal of Clinical Child and Adolescent Psychology, 37(1), 8–38.
- Volkmar, F. R., & Wiesner, L. A. (2009). Advances in the diagnosis and treatment of autism spectrum disorders. F1000 Medicine Reports, 1, 61.
- World Health Organization. (2021). Autism spectrum disorders. Retrieved from https://www.who.int/news-room/questions-and-answers/item/autism-spectrum-disorders

APPENDIX

APPENDIX

❖ TEACHER LOGIN ACTIVITY.JAVA

```
package com.example.childcare.Teacher;


import androidx.appcompat.app.AppCompatActivity;


import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.Toast;


import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.example.childcare.Config;
import com.example.childcare.R;
import com.example.childcare.TeacherRegisterActivity;
import com.example.childcare.UserHomeActivity;
```

```
import com.example.childcare.user.SessionManager;
import com.example.childcare.user.UserLoginActivity;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class TeacherLoginActivity extends AppCompatActivity {
LinearLayout lin;
    EditText ur_email,ur_password;
    Button ur_login;
    String email,pass;
    String Status,Message;
    String id,name,mobile,place,em,pwd,img,college;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_teacher_login );
        lin=findViewById( R.id.lin );
        lin.setOnClickListener( new View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

```
            startActivity( new Intent(TeacherLoginActivity.this, TeacherRegisterActivity.class)
);
        }
    } );
    ur_login=findViewById( R.id.ur_login );
    ur_email=findViewById( R.id.ur_email );
    ur_password=findViewById( R.id.ur_password );
    ur_login.setOnClickListener( new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            log();
        }

    } );

}

private void log() {
    email = ur_email.getText().toString();
    pass =ur_password.getText().toString();



    String url = Config.baseurl + "teacher_login.php";
```

```
if (TextUtils.isEmpty( email )) {
    ur_email.requestFocus();

    ur_email.setError( "Required Field" );
    return;
}
if (TextUtils.isEmpty( pass )) {
    ur_password.requestFocus();
    ur_password.setError( "Required Field" );
    return;
}



StringRequest stringRequest = new StringRequest( Request.Method.POST, url,

        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {

                    //  Toast.makeText( UserLoginActivity.this, response,
Toast.LENGTH_SHORT ).show();

                    JSONObject c = new JSONObject( response );
                    Status = c.getString( "status1" );
```

```
                Message = c.getString( "message" );


                id=c.getString( "tid" );
                name=c.getString( "name" );
                mobile=c.getString( "mobile" );
                place=c.getString( "place" );
                em=c.getString( "email" );
                pwd=c.getString( "password" );
                college=c.getString( "school" );
                img=c.getString( "status" );









                checkLogin();
            } catch (JSONException e) {
                e.printStackTrace();
            }


        }
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
```

```
                    Toast.makeText( TeacherLoginActivity.this, String.valueOf( error ),
Toast.LENGTH_SHORT ).show();
                }
            } ) {
        @Override
        protected Map<String, String> getParams() {
            Map<String, String> params = new HashMap<>();
            params.put( "email", email );
            params.put( "password", pass );
            return params;
        }
    };
    RequestQueue requestQueue = Volley.newRequestQueue( this );
    requestQueue.add( stringRequest );
  }
  private void checkLogin() {
    if (Status.equals("0")) {
        Toast.makeText( this, "Login Failed", Toast.LENGTH_SHORT ).show();
    } else {
        new TeacherSessionManager(TeacherLoginActivity.this ).createLoginSession(
id,name,mobile,place,em,pwd,college,img);
        Toast.makeText(this, "Success", Toast.LENGTH_SHORT).show();
        Intent i = new Intent(TeacherLoginActivity.this,TeacherHomeActivity.class);
        startActivity(i);
        finish();
```

```
        }


    }
}
```

❖ ANDROIDMANIFEST.XML

```
<?xml version="1.0" encoding="utf-8"?>
 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.childcare">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-feature
       android:name="android.hardware.camera"
       android:required="true" />
    <application
       android:allowBackup="true"
       android:icon="@drawable/ch"
       android:label="@string/app_name"
       android:roundIcon="@drawable/ch"
       android:supportsRtl="true"
       android:theme="@style/Theme.ChildCare"
       android:usesCleartextTraffic="true">
       <activity
```

```
    android:name=".user.LevelActivity"
    android:exported="false" />
<activity
    android:name=".user.View_point"
    android:exported="false" />
<activity
    android:name=".user.ChatActivity"
    android:exported="false" />
<activity
    android:name=".faceemr.PredictActivity"
    android:exported="false" />
<activity
    android:name=".Teacher.Point"
    android:exported="false" />
<activity
    android:name=".Teacher.View_Points"
    android:exported="false" />
<activity
    android:name=".Teacher.View_students"
    android:exported="false" />
<activity
    android:name=".Teacher.TeacherProfile"
    android:exported="false" />
<activity
    android:name=".user.ViewActivity"
```

```
    android:exported="false" />
<activity
    android:name=".user.VerifyActivity"
    android:exported="false" />
<activity
    android:name=".user.QuizActivity3"
    android:exported="false" />
<activity
    android:name=".user.QuizActivity2"
    android:exported="false" />
<activity
    android:name=".user.Alpha_details"
    android:exported="false" />
<activity
    android:name=".user.Alphabet_Puzzle"
    android:exported="false" />
<activity
    android:name=".user.Number_details"
    android:exported="false" />
<activity
    android:name=".user.Number_Puzzle"
    android:exported="false" />
<activity
    android:name=".user.Colour_detail"
    android:exported="false" />
```

```
<activity
    android:name=".user.View_Colours"
    android:exported="false" />
<activity
    android:name=".user.View_Doctors"
    android:exported="false" />
<activity
    android:name=".user.Image_Puzzle"
    android:exported="false" />
<activity
    android:name=".user.Veg_detail"
    android:exported="false" />
<activity
    android:name=".user.View_Vegetables"
    android:exported="false" />
<activity
    android:name=".user.View_Fruits"
    android:exported="false" />
<activity
    android:name=".user.Music_detail"
    android:exported="false" />
<activity
    android:name=".user.View_Music"
    android:exported="false" />
<activity
```

```
    android:name=".user.View_Stories"
    android:exported="false" />
<activity
    android:name=".Teacher.TeacherHomeActivity"
    android:exported="false" />
<activity
    android:name=".TeacherRegisterActivity"
    android:exported="false" />
<activity
    android:name=".Teacher.TeacherLoginActivity"
    android:exported="false" />
<activity
    android:name=".user.ProcessActivity"
    android:exported="false" />

<service android:name=".user.AlertService" />
<service android:name=".user.ClapService" />

<activity
    android:name=".user.Upload_music"
    android:exported="false" />
<activity
    android:name=".user.View_points"
    android:exported="false" />
<activity
```

```
        android:name=".user.Video_detail"
        android:exported="false" />
    <activity
        android:name=".UploadVideo"
        android:exported="false" />
    <activity
        android:name=".user.Quiz_detail"
        android:exported="false" />
    <activity
        android:name=".user.Puzzle_Detail"
        android:exported="false" />
    <activity
        android:name=".user.Recognize_detail"
        android:exported="false" />
    <activity
        android:name=".UserHomeActivity"
        android:exported="false" />
    <activity
        android:name=".user.UserRegistrationActivity"
        android:exported="false" />
    <activity
        android:name=".user.UserLoginActivity"
        android:exported="false" />
    <activity
        android:name=".MainActivity"
```

```
            android:exported="true">
            <intent-filter>
               <action android:name="android.intent.action.MAIN" />

               <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
         </activity>
      </application>

   </manifest>
```

- **ACTIVITY TEACHER LOGIN XML**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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=".User.UserRegistrationActivity"
   android:weightSum="8"
   android:orientation="vertical"
   android:background="@color/white"
   android:paddingLeft="30dp"
   android:paddingRight="30dp">
```

```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_centerInParent="true"
    android:paddingTop="20dp">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_marginTop="20dp"
        >
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">


            <ImageView
                android:id="@+id/unni"
                android:layout_width="178dp"
                android:layout_height="123dp"
                android:layout_centerHorizontal="true"
                android:layout_marginTop="30dp"
                android:src="@drawable/ic_login_hero" />
```

```
<TextView
    android:id="@+id/messi"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/unni"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:fontFamily="serif"
    android:text="LOGIN"
    android:textColor="@color/black"
    android:textSize="22sp"
    android:textStyle="bold" />

<LinearLayout
    android:id="@+id/mon"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/messi"
    android:layout_marginTop="40dp"
    android:orientation="vertical">


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

```
            app:startIconDrawable="@drawable/ic_baseline_email_24"
            app:startIconTint="@color/purple_200"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense">


            <EditText
                android:id="@+id/ur_email"
                android:layout_width="match_parent"
                android:layout_height="55dp"
                android:layout_marginTop="15dp"
                android:ems="10"
                android:fontFamily="serif"
                android:hint="Email"
                android:inputType="textEmailAddress"
                android:padding="20dp"
                android:textColorHint="#000" />
        </com.google.android.material.textfield.TextInputLayout>
        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:startIconDrawable="@drawable/ic_baseline_lock_24"
            app:startIconTint="@color/purple_200"
            app:passwordToggleEnabled="true"
```

```
style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense">


        <EditText
            android:id="@+id/ur_password"
            android:layout_width="match_parent"
            android:layout_height="55dp"
            android:layout_marginTop="15dp"
            android:ems="10"
            android:fontFamily="serif"
            android:hint="Password"
            android:inputType="textPassword"
            android:padding="20dp"
            android:textColorHint="#000" />
    </com.google.android.material.textfield.TextInputLayout>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="LOGIN"
        android:textColor="@color/white"
        android:layout_marginTop="20dp"
        android:layout_below="@id/mon"
        android:id="@+id/ur_login"
        android:background="@drawable/button"/>
```

```
</LinearLayout>
        <LinearLayout
            android:id="@+id/lin"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_marginTop="450dp"
            android:layout_marginLeft="70dp"
            >
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Don`t have an account?"
                android:textColor="#000"
                android:textStyle="bold"/>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Register"
                android:textColor="@color/purple_200"
                android:textStyle="bold"
                android:id="@+id/TvRegister"/>
```

```
                </LinearLayout>

            <ProgressBar
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerInParent="true"
                android:id="@+id/progress"
                android:visibility="gone"/>
        </RelativeLayout>
    </LinearLayout>
  </ScrollView>
</RelativeLayout>
```

❖ TEACHER REGISTER ACTIVITY.JAVA

```
package com.example.childcare;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
```

```
import android.widget.Toast;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.example.childcare.Teacher.TeacherLoginActivity;
import com.example.childcare.user.UserLoginActivity;
import com.example.childcare.user.UserRegistrationActivity;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.HashMap;
import java.util.Map;

public class TeacherRegisterActivity extends AppCompatActivity {
    EditText
user_name,user_email,user_place,user_password,user_phone,user_school;
    Button user_register;
    String name,email,place,password,phone,school;
    String Status,Message;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_teacher_register );
```

```
    user_name=findViewById( R.id.user_name );
    user_email=findViewById( R.id.user_email );
    user_place=findViewById( R.id.user_place );
    user_password=findViewById( R.id.user_password );
    user_phone=findViewById( R.id.user_phone );
    user_register=findViewById( R.id.user_register );
    user_school=findViewById( R.id.user_school );

    user_register.setOnClickListener( new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            register();
        }
    } );
}

private void register() {
    name = user_name.getText().toString();
    email = user_email.getText().toString();
    phone = user_phone.getText().toString();
    password =user_password.getText().toString();
    place=user_place.getText().toString();
    school=user_school.getText().toString();
```

```
String url = Config.baseurl + "teacher_register.php";


if (TextUtils.isEmpty( name )) {
    user_name.requestFocus();
    user_name.setError( "Required Field" );
    return;
}
if (TextUtils.isEmpty( email )) {
    user_email.requestFocus();
    user_email.setError( "Required Field" );
    return;
}
if (TextUtils.isEmpty( phone )) {
    user_phone.requestFocus();
    user_phone.setError( "Required Field" );
    return;
}
if (TextUtils.isEmpty( password )) {
    user_password.requestFocus();
    user_password.setError( "Required Field" );
    return;
}
if (TextUtils.isEmpty( place )) {
    user_place.requestFocus();
    user_place.setError( "Required Field" );
```

```
            return;
        }
        if (TextUtils.isEmpty( school )) {
            user_school.requestFocus();
            user_school.setError( "Required Field" );
            return;
        }
        if (password.length() < 6) {
            user_password.setError( "Password Must be 6 Characters");
            return;
        }
        if (phone.length() < 10) {
            user_phone.setError( "Enter Valid Phone Number");
            return;
        }
        if (phone.length() > 10) {
            user_phone.setError( "Enter Valid Phone Number");
            return;
        }


        StringRequest stringRequest = new StringRequest( Request.Method.POST,
    url,

                new Response.Listener<String>() {
```

```
                @Override
                public void onResponse(String response) {
                    try {
                        JSONObject c = new JSONObject( response );
                        Status = c.getString( "status" );
                        Message = c.getString( "message" );
                        checkLogin();
                    } catch (JSONException e) {
                        e.printStackTrace();
                    }

                }
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    Toast.makeText( TeacherRegisterActivity.this, String.valueOf(
error ), Toast.LENGTH_SHORT ).show();
                }
            } ) {
        @Override
        protected Map<String, String> getParams() {
            Map<String, String> params = new HashMap<>();
            params.put( "name", name );
            params.put( "mobile", phone );
```

```
                params.put( "email", email );
                params.put( "password", password );
                params.put( "place",place );
                params.put( "school",school );
                return params;
            }
        };
        RequestQueue requestQueue = Volley.newRequestQueue( this );
        requestQueue.add( stringRequest );
    }
    private void checkLogin() {
        if (Status.equals("0")) {
            Toast.makeText( this, "Registration Failed", Toast.LENGTH_SHORT
).show();
        } else {
            Toast.makeText(this, "Registerd successfully",
Toast.LENGTH_SHORT).show();
            Intent i = new Intent(TeacherRegisterActivity.this,
TeacherLoginActivity.class);
            startActivity(i);
            finish();
        }


    }
```

```
    }
```

❖ ANDROIDMANIFEST.XML

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.childcare">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-feature
        android:name="android.hardware.camera"
        android:required="true" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/ch"
        android:label="@string/app_name"
        android:roundIcon="@drawable/ch"
        android:supportsRtl="true"
        android:theme="@style/Theme.ChildCare"
        android:usesCleartextTraffic="true">
        <activity
            android:name=".user.LevelActivity"
            android:exported="false" />
        <activity
            android:name=".user.View_point"
```

```
    android:exported="false" />
<activity
    android:name=".user.ChatActivity"
    android:exported="false" />
<activity
    android:name=".faceemr.PredictActivity"
    android:exported="false" />
<activity
    android:name=".Teacher.Point"
    android:exported="false" />
<activity
    android:name=".Teacher.View_Points"
    android:exported="false" />
<activity
    android:name=".Teacher.View_students"
    android:exported="false" />
<activity
    android:name=".Teacher.TeacherProfile"
    android:exported="false" />
<activity
    android:name=".user.ViewActivity"
    android:exported="false" />
<activity
    android:name=".user.VerifyActivity"
    android:exported="false" />
```

```
<activity
    android:name=".user.QuizActivity3"
    android:exported="false" />
<activity
    android:name=".user.QuizActivity2"
    android:exported="false" />
<activity
    android:name=".user.Alpha_details"
    android:exported="false" />
<activity
    android:name=".user.Alphabet_Puzzle"
    android:exported="false" />
<activity
    android:name=".user.Number_details"
    android:exported="false" />
<activity
    android:name=".user.Number_Puzzle"
    android:exported="false" />
<activity
    android:name=".user.Colour_detail"
    android:exported="false" />
<activity
    android:name=".user.View_Colours"
    android:exported="false" />
<activity
```

```
    android:name=".user.View_Doctors"
    android:exported="false" />
<activity
    android:name=".user.Image_Puzzle"
    android:exported="false" />
<activity
    android:name=".user.Veg_detail"
    android:exported="false" />
<activity
    android:name=".user.View_Vegetables"
    android:exported="false" />
<activity
    android:name=".user.View_Fruits"
    android:exported="false" />
<activity
    android:name=".user.Music_detail"
    android:exported="false" />
<activity
    android:name=".user.View_Music"
    android:exported="false" />
<activity
    android:name=".user.View_Stories"
    android:exported="false" />
<activity
    android:name=".Teacher.TeacherHomeActivity"
```

```
    android:exported="false" />
<activity
    android:name=".TeacherRegisterActivity"
    android:exported="false" />
<activity
    android:name=".Teacher.TeacherLoginActivity"
    android:exported="false" />
<activity
    android:name=".user.ProcessActivity"
    android:exported="false" />

<service android:name=".user.AlertService" />
<service android:name=".user.ClapService" />

<activity
    android:name=".user.Upload_music"
    android:exported="false" />
<activity
    android:name=".user.View_points"
    android:exported="false" />
<activity
    android:name=".user.Video_detail"
    android:exported="false" />
<activity
    android:name=".UploadVideo"
```

```
    android:exported="false" />
<activity
    android:name=".user.Quiz_detail"
    android:exported="false" />
<activity
    android:name=".user.Puzzle_Detail"
    android:exported="false" />
<activity
    android:name=".user.Recognize_detail"
    android:exported="false" />
<activity
    android:name=".UserHomeActivity"
    android:exported="false" />
<activity
    android:name=".user.UserRegistrationActivity"
    android:exported="false" />
<activity
    android:name=".user.UserLoginActivity"
    android:exported="false" />
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
```

```
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>
```

❖ ACTIVITY TEACHERLOGIN.XML

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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=".User.UserRegistrationActivity"
    android:weightSum="8"
    android:orientation="vertical"
    android:background="@color/white"
    android:paddingLeft="30dp"
    android:paddingRight="30dp">
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
```

```
android:layout_alignParentLeft="true"
android:layout_centerInParent="true"
android:paddingTop="20dp">
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:orientation="vertical"
   android:layout_marginTop="20dp"
   >
   <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content">

      <ImageView
         android:id="@+id/unni"
         android:layout_width="178dp"
         android:layout_height="123dp"
         android:layout_centerHorizontal="true"
         android:layout_marginTop="30dp"
         android:src="@drawable/ic_login_hero" />

      <TextView
         android:id="@+id/messi"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
```

```
    android:layout_below="@id/unni"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:fontFamily="serif"
    android:text="LOGIN"
    android:textColor="@color/black"
    android:textSize="22sp"
    android:textStyle="bold" />

<LinearLayout
    android:id="@+id/mon"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/messi"
    android:layout_marginTop="40dp"
    android:orientation="vertical">

    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:startIconDrawable="@drawable/ic_baseline_email_24"
        app:startIconTint="@color/purple_200"
```

```
style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense
">


        <EditText
            android:id="@+id/ur_email"
            android:layout_width="match_parent"
            android:layout_height="55dp"
            android:layout_marginTop="15dp"
            android:ems="10"
            android:fontFamily="serif"
            android:hint="Email"
            android:inputType="textEmailAddress"
            android:padding="20dp"
            android:textColorHint="#000" />
    </com.google.android.material.textfield.TextInputLayout>
    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:startIconDrawable="@drawable/ic_baseline_lock_24"
        app:startIconTint="@color/purple_200"
        app:passwordToggleEnabled="true"
```

```
style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense
">

        <EditText
            android:id="@+id/ur_password"
            android:layout_width="match_parent"
            android:layout_height="55dp"
            android:layout_marginTop="15dp"
            android:ems="10"
            android:fontFamily="serif"
            android:hint="Password"
            android:inputType="textPassword"
            android:padding="20dp"
            android:textColorHint="#000" />
    </com.google.android.material.textfield.TextInputLayout>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="LOGIN"
        android:textColor="@color/white"
        android:layout_marginTop="20dp"
        android:layout_below="@id/mon"
        android:id="@+id/ur_login"
```

```
        android:background="@drawable/button"/>



    </LinearLayout>
        <LinearLayout
            android:id="@+id/lin"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_marginTop="450dp"
            android:layout_marginLeft="70dp"
            >
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Don`t have an account?"
                android:textColor="#000"
                android:textStyle="bold"/>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Register"
                android:textColor="@color/purple_200"
                android:textStyle="bold"
                android:id="@+id/TvRegister"/>
```

```
                </LinearLayout>

            <ProgressBar
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerInParent="true"
                android:id="@+id/progress"
                android:visibility="gone"/>
        </RelativeLayout>
    </LinearLayout>
</ScrollView>
</RelativeLayout>
```