**1. INTRODUCTION**

Edventure is an innovative educational gaming app designed to make learning fun and engaging for children. The platform provides interactive and gamified educational experiences that enhance cognitive skills, problem-solving abilities, and creativity. By blending entertainment with education, Edventure ensures that children stay motivated while acquiring essential knowledge across various subjects.

The Edventure app includes a rich collection of games and activities that cater to different age groups and learning levels. These activities are designed by educators and child development experts to ensure that children receive the best learning experience while enjoying themselves. The app integrates progress tracking, adaptive learning algorithms, and parental controls to provide a tailored experience for each child.

For administrators and educators, the platform offers a web-based interface to manage content, track student progress, and customize learning paths. Parents can monitor their child’s development, set learning goals, and receive insights into their progress through an intuitive dashboard.

Edventure is more than just a gaming app; it is a revolutionary approach to modern education that fosters learning through play, making education more interactive, accessible, and enjoyable for young learners.

**2. SYSTEM ANALYSIS**

System analysis is a step-by-step process used to identify and develop or acquire the software need to control the processing of specific application. System analysis is a continuing activity the stages of the systems development. System analysis is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system. The outputs from the organization are traced through the various processing that the input phases through in the organization. This involves gathering information and using structured tools for analysis. A detailed study of this process must be made by various techniques like interviews, questionnaires etc.

**2.1 EXISTING SYSTEM**

Traditional learning methods often fail to capture children's attention, making education seem monotonous and less effective. Many existing educational apps lack engaging game mechanics, leading to decreased interest and motivation among young learners. Additionally, most platforms do not offer real-time progress tracking or adaptive learning experiences tailored to each child's needs.

**Disadvantages of the Existing System:**

* Lack of engagement due to traditional and passive learning methods.
* Limited customization and adaptability to individual learning styles.
* Absence of real-time progress tracking for parents and educators.
* Poor integration of interactive and gamified elements in learning applications.

**2.2 PROPOSED SYSTEM**

The proposed system, Edventure, introduces a fully interactive educational gaming platform that combines entertainment with structured learning. It ensures that children remain engaged while progressing through various educational activities designed to boost their knowledge and skills.

**Advantages of Edventure:**

* **Gamified Learning:** Engaging games and challenges make learning enjoyable.
* **Personalized Experience:** AI-driven adaptive learning ensures that content matches the child's pace and abilities.
* **Real-Time Tracking:** Parents and educators can monitor progress and adjust learning plans accordingly.
* **Multi-Subject Coverage:** Includes topics such as mathematics, science, language, and creativity-enhancing activities.
* **Interactive and Reward-Based System:** Encourages motivation and active participation.

**2.3 SYSTEM REQUIREMENT SPECIFICATION**

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform. An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-worked situations.

**Problem to be Solved**

This system solves the lack of engagement and manual effort involved in traditional learning methods for kids. It replaces boring study routines with interactive game-based learning and provides a platform for parents to track their child’s educational progress easily.

**Customer Requirements**

* The system should be fast and responsive.
* User-friendly interface for kids, parents, and admin.
* Secure storage and management of user data.
* Efficient tracking of learning progress and performance.
* Engaging content that motivates kids to learn through play.

**What the Developer Needs to Know**

* Must understand the drawbacks of traditional learning systems.
* Must know the needs of children, parents, and educational goals for the proposed system.
* Must design with a focus on user-friendly design and child safety.

**Business Requirements**

The system should be feasible for both developers and clients. It should be completed within the given timeline with an effective and scalable solution. Developers should handle the development, deployment, updates, and provide necessary user guidance or training for parents and admins to use the app.

**User Requirements**

This section defines the requirements expected by the users of the Edventure system:

* Admin has full control over managing content, users, and monitoring platform activities.
* Parents can view and track their child's progress and set learning goals.
* Children can play subject-based educational games, view scores, and earn rewards.
* The system should provide quick navigation and interactive interfaces suitable for kids.

**Functional Requirements**

Functional requirements specify the features and actions the Edventure system should perform:

* Login – Login functionality for admin, parents, and children.
* Registration – Parents can register themselves and create a profile for their child.
* Edit Profile – Users (parents) can update their profile and child's details.
* View Progress – Parents can view learning progress, achievements, and performance reports of their child.
* Play Games – Children can play educational games based on subjects and levels.
* Leaderboard – Displays top-performing kids based on their scores.
* Logout – All users can securely logout from the app.

**2.3.1 HARDWARE SPECIFICATION**

Processor : Intel core i5

Speed : 2.50GHz

System bus : 64 bits

Memory : 16GB RAM

Hard disk : 40GB or higher

Monitor : 15.6-inch, Anti-Glare

Keyboard : 100 keys

Pointing device : Touchpad

**2.3.2 SOFTWARE SPECIFICATION**

Operating System : Windows

Front End : Flutter (Dart)

Back End : Supabase

Development Tools : Android Studio , Visual Studio Code

Platform : Android , Web

Browser Compatibility : Google Chrome

**2.3.3 FRONT END**

**FLUTTER**

Flutter is an open-source UI software development kit created by Google. It is used to develop cross-platform applications from a single codebase, allowing developers to build high-performance, visually appealing apps for mobile, web, and desktop.

* Flutter uses the Dart programming language, which is optimized for fast app development.
* It provides a rich set of pre-designed widgets that help create responsive and attractive user interfaces.
* Flutter’s hot reload feature allows developers to see real-time updates without restarting the application.
* It ensures a smooth user experience with a high frame rate and native-like performance.

**Common Uses of Flutter**

* Flutter is used for building mobile applications that work on both Android and iOS.
* It enables web and desktop development using the same codebase.
* It is ideal for applications requiring rich UI elements and animations.
* It integrates seamlessly with Supabase for backend services such as authentication, cloud storage, and real-time databases.

**Characteristics of Flutter**

Flutter's advantages include:

* Cross-Platform Development – Develop once, deploy anywhere.
* Fast Development – Hot reload accelerates the development process.
* Beautiful UI – Uses Material Design and Cupertino widgets.
* High Performance – Delivers near-native performance.
* Strong Community Support – Backed by Google and a vast developer community

**2.3.4 BACK END**

**SUPABASE**

Supabase is an open-source backend-as-a-service (BaaS) platform that provides a scalable and powerful alternative to Firebase. It is built on PostgreSQL and offers real-time capabilities, authentication, and cloud storage.

* Supabase Authentication allows secure user login using email, Google, and other OAuth providers.
* Supabase Database is a managed PostgreSQL database with real-time capabilities.
* Supabase Storage enables file and media storage with easy access control.
* Supabase Functions provide serverless computing for custom backend logic.
* Supabase API allows seamless integration with Flutter applications

**2.4 FEASIBILITY ANALYSIS**

A feasibility study is an evaluation and analysis of the potential of the proposed project which is based on extensive investigation and research to give full comfort to the decision makers. Feasibility studies aim to objectively and rationally uncover the strength and weakness of existing business of proposed venture, opportunities and threads as presented by the environment, the resources required to carry through, and ultimately the process for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to attain. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations.The four aspects in the feasibility study are:

* Technical feasibility
* Economic feasibility
* Operational feasibility
* Behavioural feasibility

**Technical Feasibility**

The technical feasibility centres on the existing system and what extend it can support the proposed addition. The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. The minimum requirements of the system are met by average user. The developer system has á modest technical requirement as only minimal or null changes are required for implementing system.Normally associated with the technical feasibility includes:

* Development risk
* Resource availability
* Technology

The proposed system can work without any additional hardware or software support other than the computer system and networks. So, I analysed that the proposed system is much more technically feasible than other systems when comparing with the benefits of the new system.

**Economic Feasibility**

Economic feasibility analysis is also known as cost/benefit analysis. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. The proposed system reduces the operating cost in terms of time by automating the process. This system is economically feasible.

**Operational Feasibility**

Operational feasibility is a measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

**Behavioural Feasibility**

People are inherently resistant to changes and computer is known for facilitating the chances. An estimate should be made to how strongly the users react towards the e development of the system. The proposed system consumes less time. Thus, the people are made to engage in some other important work.

**2.5 DATA FLOW DIAGRAM (DFD)**

**2.5.1 INTRODUCTION TO DATA FLOW DIAGRAM**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing (structured design).

Data flow diagrams were invented by Larry Constantine, the original developer of structured design based on Martin and Estrin's "data flow graph" model of computation.

Data flow diagrams (DFDs) are one of the three essential perspectives of Structured System Analysis and Design Method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users can visualize how the system will operate, what the system will accomplish. and how the system will be implemented. The old system's data flow diagrams can be drawn up and compared with the new system's data flow diagrams to drawn comparisons to implement a more efficient system. Data flow diagrams can be used to provide the end user with physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data flow diagram.

Developing a data flow diagram helps in identifying the transaction data in the data model. There are different notations to draw data flow diagrams, defining different visual representation for process, data stores, data flow, and external entities. The first step is to draw a data flow diagram (DFD). A DFD also known as "bubble chart" has the purpose of clarifying system requirements and identifying major transformation that will become program in system design. So, it is starting point of the design phase that functionally decompose the requirements specification down to the lowest level of details DFD consists of series of bubbles joined by lines. The bubbles represent data transformation and the Iines represent data flow in the system.

**DFD Symbols**

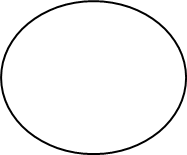
* Square- Defines source or destination of system.



* Data flow - Identifies data flow Circle.



* Bubble - Represents a process that transforms incoming data to outgoing data.



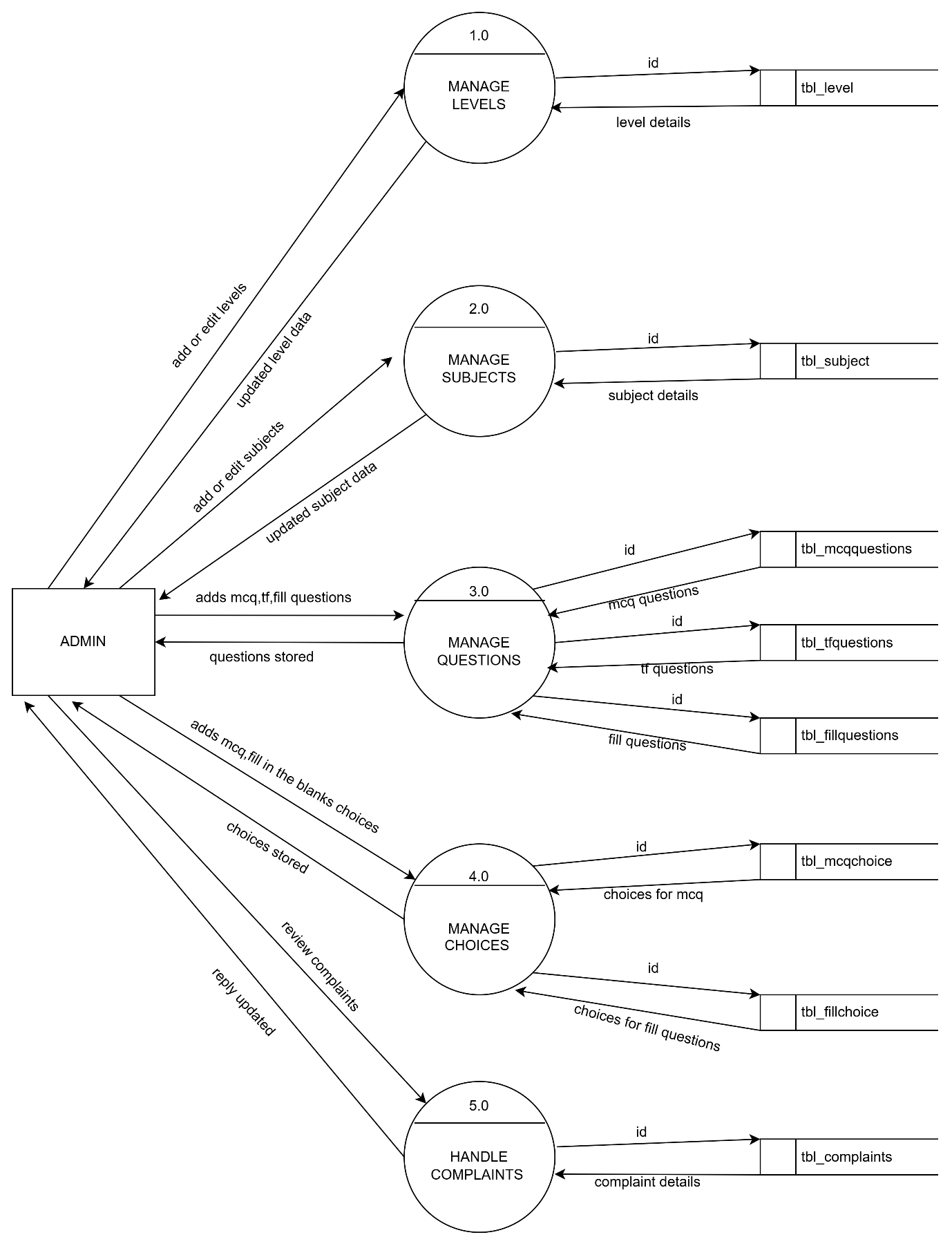
* Open rectangle - Data store

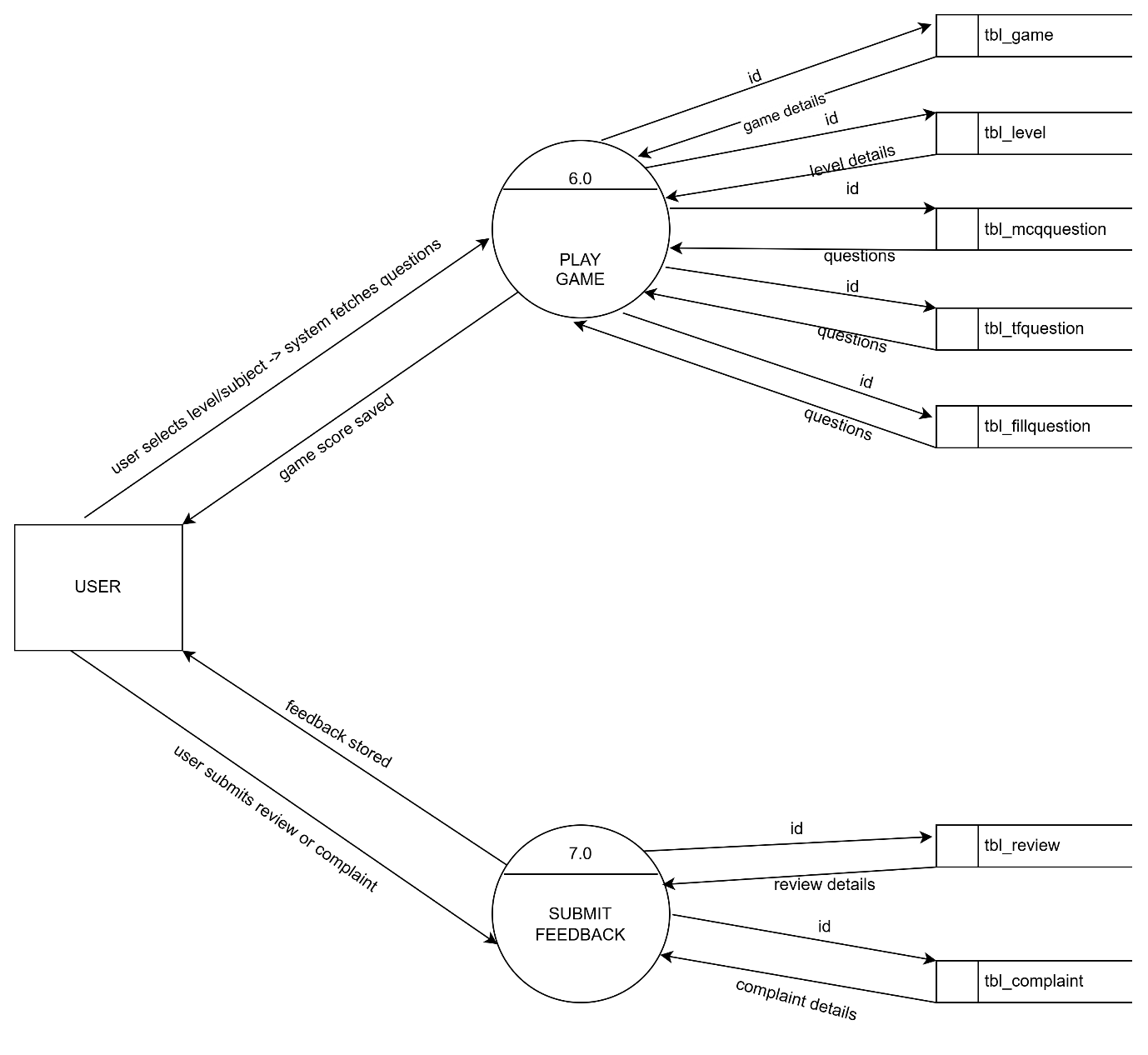
|  |  |
| --- | --- |
|  |  |

**LEVEL – 0**



**LEVEL – 1**





**3. SYSTEM DESIGN**

**3.1 INPUT DESIGN**

Input design ensures that users interact with the system effectively, minimizing errors and ensuring valid data entry. InEdventure, inputs are designed to be user-friendly and suitable for children, with proper validation and error handling.

Inputs are collected through interactive components like:

* Buttons
* Selection Menus
* Rating Bars
* Text Fields (with validation)

**User Roles & Input Responsibilities:**

| **User Role** | **Input Responsibilities** |
| --- | --- |
| **Admin** | - Manages users (students & parents). - Adds and manages levels, questions, subjects. - Manages feedback, complaints, and leaderboard. |
| **User (Student/Player)** | - Plays levels (MCQ, TF, Fill in the Blanks). - Provides feedback for levels. - Can send complaints for issues. |
| **Parent** | - Views child's progress via reports or dashboards (Optional). |

**3.2 OUTPUT DESIGN**

Output design focuses on delivering structured and clear information to the users. Outputs in Edventure are made visually attractive and engaging for children.

**Types of Outputs in Eduplay Adventures:**

| **Output Type** | **Description** |
| --- | --- |
| **Score Display** | Displays score after completing each level. |
| **Leaderboard** | Shows the top scorers across all levels. |
| **Feedback Acknowledgement** | Displays confirmation after review submission. |
| **Complaints Reply** | Users receive a reply to their complaint from the admin. |
| **Progress Reports** | Shows level completion status and achievements. |

**3.3 DATABASE DESIGN**

The Edventure database is designed for optimized storage, retrieval, and security of user and game data. It uses normalization, foreign key constraints, and indexing for performance enhancement.

**Key Steps Followed in Database Design:**

1. Identifying data entities like Users, Levels, Subjects, Questions, Scores, Reviews, etc.
2. Designing Entity-Relationship Diagrams (ERD) to map relationships.
3. Defining tables with Primary Keys and Foreign Keys.
4. Ensuring data integrity through validation.
5. Optimizing queries for faster retrieval in a gaming environment.

**Database Tables in Edventure:**

| **Table Name** | **Purpose** |
| --- | --- |
| **tbl\_admin** | Stores admin details (admin\_id, name, email, password). |
| **tbl\_user** | Stores user details including parent information and date of birth. |
| **tbl\_level** | Stores details of each level (level\_id, name, time duration). |
| **tbl\_subject** | Stores subject names used in the game. |
| **tbl\_game** | Stores game results (game\_id, score, question count, type of game). |
| **tbl\_mcqquestion** | Stores MCQ questions with optional images. |
| **tbl\_tfquestion** | Stores True/False type questions. |
| **tbl\_fillquestion** | Stores Fill in the Blank type questions. |
| **tbl\_choice** | Stores options for MCQ questions and correct answer indicator. |
| **tbl\_addfillchoice** | Stores options for Fill in the Blank type questions. |
| **tbl\_review** | Stores feedback from users after completing levels (rating, content, date). |
| **tbl\_complaint** | Stores user complaints along with admin reply. |

This well-structured database ensures efficient data management, enhances performance, and provides a smooth learning and gaming experience for children in Edventure.

The various database tables that are used in this project are the following:

1. **tbl\_admin**

Primary key : admin\_id

Description : Stores the details of admin.

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **DESCRIPTION** |
| admin\_id | UUID | Unique id of admin |
| admin\_name | TEXT | Name of admin |
| admin\_email | TEXT | Email id of admin |
| admin\_password | TEXT | Password of admin |

1. **tbl\_level**

Primary key : level\_id

Description : Stores the details of levels.

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **DESCRIPTION** |
| level\_id | INT | Unique id of level |
| level\_name | TEXT | Name of the level |
| level\_time | INT | Time in seconds for each level |

1. **tbl\_user**

Primary key : user\_id

Description : Stores the details of users.

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **DESCRIPTION** |
| user\_id | UUID | Unique id of user |
| user\_name | TEXT | Name of the user |
| parent \_name | TEXT | Name of the parent of the user |
| parent\_email | TEXT | Email id of the parent |
| user\_password | TEXT | Password of the user |
| user\_dob | TEXT | Date of birth of the user |

1. **tbl\_game**

Primary key : game\_id

Foreign key : user\_id , level\_id , subject\_id

Description : Stores the details of game.

|  |  |  |
| --- | --- | --- |
| **FILED NAME** | **DATATYPE** | **DESCRIPTION** |
| game\_id | INT | Unique id of game |
| qstn\_level | INT | Level of questions |
| game\_score | INT | Score of the game |
| qstn\_count | INT | Count of the question |
| game\_type | TEXT | Type of game |

1. **tbl\_review**

Primary key : review\_id

Foreign key : user\_id

Description : Stores the details of review.

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **DESCRIPTION** |
| review\_id | INT | Unique id of review |
| review\_rating | TEXT | Rating of review |
| review\_content | TEXT | Content of review |
| review\_date | TEXT | Date of review |

1. **tbl\_subject**

Primary key : subject\_id

Description : Stores the name of subjects.

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **DESCRIPTION** |
| subject\_id | INT | Unique id of subject |
| subject\_name | TEXT | Name of the subject |

1. **tbl\_choice**

Primary key : choice\_id

Foreign key : question\_id

Description : Stores the details of choices.

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **DESCRIPTION** |
| choice\_id | INT | Unique id of choice |
| answer | TEXT | Answer to each question as four options |
| is\_correct | BOOL | State whether the answers is correct or incorrect |

1. **tbl\_addfillchoice**

Primary key : fillchoice\_id

Foreign key : fillquestion\_id

Description : Stores the details of choices.

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **DESCRIPTION** |
| choice\_id | INT | Unique id of choice |
| answer | TEXT | Answer to each question as four options |
| is\_correct | BOOL | State whether the answers is correct or incorrect |

1. **tbl\_complaint**

Primary key : complaint\_id

Foreign key : user\_id

Description : Stores the details of complaint.

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **DESCRIPTION** |
| complaint\_id | INT | Unique id of complaint |
| complaint\_status | INT | Status of complaint |
| complaint\_text | TEXT | Content of complaint |
| complaint\_reply | TEXT | Reply to the complaint |

1. **tbl\_fillquestion**

Primary key : fillquestion\_id

Foreign key : level\_id , subject\_id

Description : Stores the details of fill questions.

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **DESCRIPTION** |
| fillquestion\_id | INT | Unique id of fillquestion |
| qstn\_text1 | TEXT | Text before blank |
| qstn\_text2 | TEXT | Text after blank |
| qstn\_level | INT | Level of question set |

1. **tbl\_mcqquestion**

Primary key : mcqquestion\_id

Foreign key : level\_id , subject\_id

Description : Stores the details of multiple choice questions.

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **DESCRIPTION** |
| mcqquestion\_id | INT | Unique id of mcq question |
| question | TEXT | Main question |
| sub\_question | TEXT | Sub question for main question |
| question\_level | INT | Level of question set |
| image | TEXT | Image which relates to question or answer |

1. **tbl\_tfquestion**

Primary key : tfquestion\_id

Foreign key : subject\_id , level\_id

Description : Stores the details of true or false questions.

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **DESCRIPTION** |
| tfquestion\_id | INT | Unique id of tfquestion |
| tfquestion\_text | TEXT | True or false question |
| question\_file | TEXT | Image if needed to show |
| question\_level | INT | Level of question set |
| question\_iscorrect | BOOL | Whether the answer is true or false |

**4. SYSTEM IMPLEMENTATION AND TESTING**

**4.1 SYSTEM TESTING**

Testing is the process of examining the software to compare the actual behaviour with that of the excepted behaviour. The major goal of software testing is to demonstrate that faults are not present. In order to achieve this goal, the tester executes the program with the intent of finding errors. Though testing cannot show absence of errors but by not showing their presence it is considered that these are not present. System testing is defined as the process by which one detects the defects in the software. Any software development organization or team has to perform several processes. Software testing is one among them. It is the final opportunity of any programmer to detect and rectify any defects that may have appeared during the software development stage. Testing is a process of testing a program with the explicit intention of finding errors that makes the program fail. In short system testing and quality assurance is a review in software products and related documentation for completion, correctness, reliability and maintainability.

System testing is the first stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct and the goal will be successfully achieved. A series of testing are performed for the proposed system before the proposed system is ready for user acceptance testing. The testing steps are :-

1. Unit testing
2. Integration testing
3. Acceptance Testing
4. Validation
5. Output testing

System Testing provides the file assurance that software once validated mast combined with all other system elements. System testing verifies whether all elements have been combined properly and that overall system function and performance is achieved. FA the integration of modules, the validation test was carried out over the system. lt was that all the modules work well together and meet the overall system function and performance.

**1.Unit Testing**

Unit testing is caried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate errors. This has enabled the detection of errors in coding and logic.

Various test cases are prepared. For each module these test cases are implemented, and it is checked whether the module is executed as per the requirements and outputs the desired result.

In this test each service input and output parameters are checked.

In unit testing :-

* Module interface was tested to ensure that information properly flows into and out of the program under test.
* Boundary condition was tested to ensure that module operates properly at boundaries established to limit or restrict processing.
* All independent paths through the control structures were executed to ensure that all statements in the modules have been executed at least once.
* Error handling paths were also tested.

**2.Integration Testing**

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing.

Unit tested module were taken and a single program structure was built that has been dictated by the design. Incremental integration has been adopted here.

The modules are tested separately lor accuracy and modules are integrated too.th tn. using bottom-up integration i.e., by integrating from moving from bottom to the top of the system is checked and errors found during integration are rectified. In this testing individual modules were combined and he m0duie wise Shifting was verified to be alright.

The entire software was developed and tested in small segments, where errors were easy to locate and rectify. Program builds (group of modules) were constructed corresponding to the successful testing of user interaction, data manipulation analysis, and display processing and database management.

**3.Validation Testing**

Validation testing is done to ensure complete assembly of the error-free software. Validation can be termed successful only if it functions in manner. Reasonably expected by the student under validation is alpha and beta testing. The student-side validation is done in this testing phase. It is checked whether the data passed to each student is valid or not. Entering incorrect values does the validation testing and it is checked whether the errors are being considered. Incorrect values are to be discarded. The errors are rectified.

In “Pharmacie" verifications are done correctly. So, there is no chance for users to enter incorrect values. It will give error messages by using different validations. The validation testing is done very clearly and found it is error free.

**4.Output Testing**

After performing the validation testing the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format.

The output format on the screen was found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also, the output comes out as specified requirement by the user. Hence output testing does not result in any Correction in the system.

Output This project is developed based on the user choice. It is user friendly. The output format is very clear to user. Output testing is done on Pharmacy Automation System correctly.

**5.Acceptance Testing**

Acceptance involves running a suite of tests on the completed system. Each individual test, known as a Case, exercise particular operating condition of the operating condition of the user's environment or feature of the system, and will result in a pass fail, or Boolean outcome.

**4.2 SYSTEM IMPLEMENTATION**

The implementation is the final state, and it is an important phase. It involves the invalid programming system testing. user training and the operational running of developed proposed system that constitutes the application subsystems. A major task of preparing for implementation is education of users. which should really have been taken place much carrier in the project when they were belong involved in the investigation and design work. During the implementation phase system take physical shape. In order to develop a system implemented planning is very essential.

The implementation phase of the software development is concerned with translating design specification into source code. The user tests the developed system and changes are made according to their needs. Our system has been successfully implemented.

Before implementation several tests have been conducted to ensure that no errors are encountered during the operation. The implementation phase ends with an evaluation of the system after placing into the operation for a period of time.

The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from old system to new system. The system can be implemented only after testing is done and is found to be working to specifications. The implementation stage is a systems project in its own right.

The implementation stage involves following tasks:

* Careful planning.
* Investigation of system and constraints.
* Design of method to achieve change over.
* Evaluation of the changeover method.

In the case of this project all the screens are designed first. For making it to be executable, codes are written on each screen and performs the implementation by creating the database and connecting to the server. After that the system, is Checked, whether it performs all the transactions correctly. Then databases are cleared and made it to be usable to the technicians.

**Implementation Plans**

The following are the steps involved in the implementation plan of "Edventure Educational Gaming App":

* Test the system with sample users and demo data to ensure proper functionality.
* Detect and correct any errors or bugs found during testing.
* Make necessary changes and improvements based on user feedback.
* Verify the existing system features and ensure all modules work as expected.
* Install the required software utilities, set up the backend database (Supabase), and deploy the application on mobile platforms (Android & iOS).
* Provide training and guidance for parents, children, and admin users to ensure smooth usage of the application.

**5. SECURITY TECHNOLOGIES & POLICIES**

The protection of computer-based resources that includes hardware, software, data procedures and people against unauthorized use or natural disaster is known as System Security.

System Security can be divided into four related issues :-

* Security
* Integrity
* Privacy
* Confidentiality

**SYSTEM SECURITY** refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

**DATA SECURITY** is the protection of data from loss, disclosure, modification and destruction.

**SYSTEM INTEGRITY** refers to the power functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

**PRIVACY** defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

**CONFIDENTIALITY** is a special status given to sensitive information in a database to minimize the possible invasion of privacy. lt is an attribute of information that characterizes its need for protection.

**SECURITY IN SOFTWARE**

System security refers to various validations on data in form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered, and only valid operations are performed on the system. The system employees two types check and controls:

**CLIENT-SIDE VALIDATION**

Various client-side validations are used to ensure on the client side that only valid data is entered. Client-side validation saves server time and load to handle invalid data. Some checks imposed are:

* Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.
* Tab-indexes are set according to the need and taking into account the ease of user while working with the system.

**SERVER-SIDE VALIDATION**

Some checks cannot be applied at client side. Server-side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server-side checks imposed is:

* Server-side constraint has been imposed to check for the validity of primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results into a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values.
* User is intimating through appropriate messages about the successful operations or exceptions occurring at server side.
* Various Access Control Mechanisms have been built so that one user may not agitate upon another. Access permissions to various types of users are controlled according to the organizational structure. Only permitted users can log on to the system and can have access according to their category. User- name, passwords and permissions are controlled o the server side.
* Using server-side validation, constraints on several restricted operations are imposed.

**6. MAINTENANCE**

Software maintenance is the modification of a software product and delivery to correct faults, to improve performance or other attributes. Maintenance is the ease with which a program can be corrected if any error is encountered, adapted if its environment changes or enhanced if the customer desires a change in requirement. Maintenance follows conversation to extend that changes are necessary to maintain satisfactory operations relative to changes in the user's environment.

Maintenance often includes minor enhancements or corrections to problems that surface in the system's operation. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

**CATEGORIES OF MAINTENANCE**

**Corrective Maintenance**

Corrective maintenance is the most used maintenance approach, but it is easy to see its limitations. When equipment fails, it often leads to downtime in production, and sometimes damages other parts. In most cases, this is expensive. Also, if the equipment needs to be replaced, the cost of replacing it alone can be substantial. Reliability of systems maintained by this type of maintenance is unknown and cannot be measured. Corrective maintenance is possible since the consequences of failure or wearing out are not significant and the cost of this maintenance is not great.

**Perfective Maintenance**

Modification of a software product alter delivery to improve performance or maintainability. This term is used to describe changes undertaken to expand the existing requirements of the system. A successful piece or software lends to be subjected to the Succession of changes resulting in an increase in us requirements. This is based on premise that as the software becomes useful, the user experiment with new cases beyond the of Scope for which it was initially developed. Vxpansi01 requirements can take the form enhancement of existing system functionality and improvement in computational efficiency

**Adaptive Maintenance**

Modification of a software product performed after delivery to keep a product usable, changed or changing environment. Adaptive maintenance includes any work initiated because of moving the software to a different hardware or software platform. It is a change driven by the need to accommodate modifications in the environment of software system. The environment in this context refers to the totality of all conditions and influences which act from outside upon the system. A change to the whole or part of this environment will Warrant a corresponding modification of the software.

**Preventive Maintenance**

Preventive maintenance is a schedule of planned maintenance actions aimed at the prevention of breakdowns and failures. The primary goal of preventive maintenance is to prevent the failure of equipment before it occurs. It is designed to preserve and enhance equipment reliability by replacing worn components before they fail. Preventive maintenance activities include equipment checks, partial or complete overhauls at specified periods.

Long-term benefits of preventive maintenance include :-

* Improved system reliability.
* Decreased cost of replacement.
* Decreased system downtime.

**7. SCOPE FOR FUTURE ENHANCEMENT**

In the future, the Edventure educational gaming app can be enhanced by integrating advanced technologies to provide a smarter and more personalized learning experience for children. Artificial Intelligence (AI) can be incorporated to analyze a child’s learning behavior and suggest customized learning paths or content based on their strengths and weaknesses. Augmented Reality (AR) and Virtual Reality (VR) can be introduced to offer immersive and interactive educational experiences, making learning more fun and engaging. Multiplayer learning modes can be added to promote collaborative learning and friendly competition among kids. The integration of voice recognition can help children interact with the app through voice commands, making the learning process more accessible and interactive. Blockchain technology can be used to securely store children's learning records and achievements. Additionally, Edventure can include multilingual support to cater to kids from different linguistic backgrounds, and accessibility features to support children with special needs. These future enhancements will further transform Edventure into a highly advanced and inclusive educational platform.

**8. CONCLUSION**

In conclusion, Edventure is a modern educational gaming app that transforms traditional learning methods into fun-filled, interactive experiences for children. It bridges the gap between education and entertainment by providing subject-based games, quizzes, and challenges that promote learning in an engaging way. The app effectively addresses the drawbacks of conventional learning systems by providing real-time progress tracking, personalized learning paths, and a safe, user-friendly environment for kids. Future enhancements such as AI-based learning suggestions, AR/VR interactive content, voice-enabled features, and blockchain security will further elevate Edventure’s capabilities. These advancements will ensure that Edventure continues to offer a smart, secure, and enjoyable learning experience for children, while also providing parents and educators with powerful tools to guide and monitor their child’s learning journey. Edventure is poised to become an essential educational platform for kids, shaping the future of learning in a playful and innovative way.

**9. BIBLIOGRAPHY**

**BOOKS**:

* Marco L. Napoli, *Beginning Flutter: A Hands-On Guide to App Development*
* Eric Windmill, *Flutter in Action*
* Marco L. Napoli, *Mastering Dart: Advanced Techniques for Modern Applications*
* Slobodan Stojanović, *Serverless Applications with Node.js*
* Shyam Seshadri, *Building Scalable Apps with Firebase*
* Elias M. Awad, *System Analysis and Design*

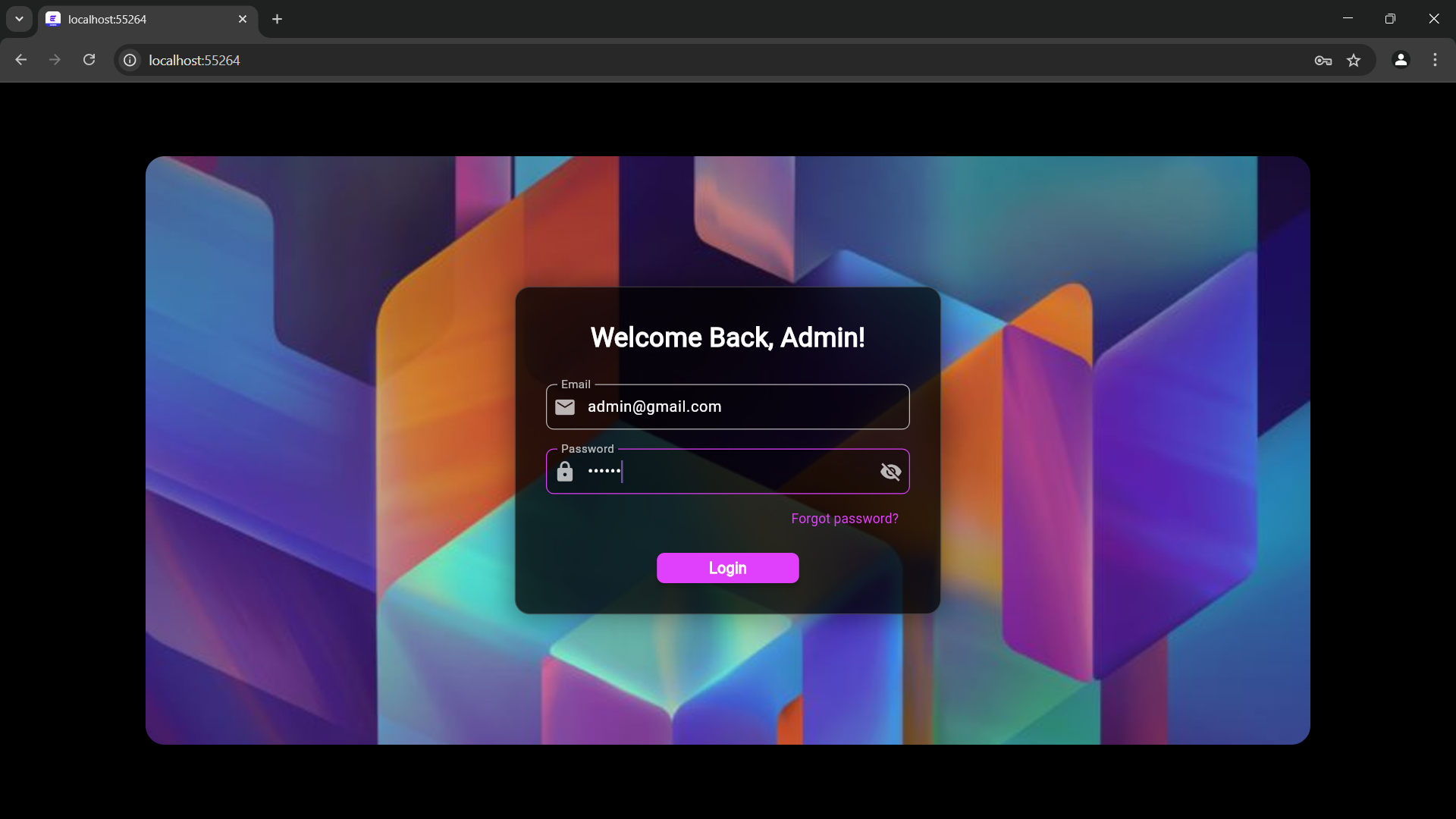
**Websites:**

* <https://flutter.dev>
* <https://dart.dev>
* <https://firebase.google.com>
* <https://supabase.com>
* <https://flutterawesome.com>
* <https://developer.android.com>
* <https://medium.com>
* <https://stackoverflow.com>
* <https://pub.dev>

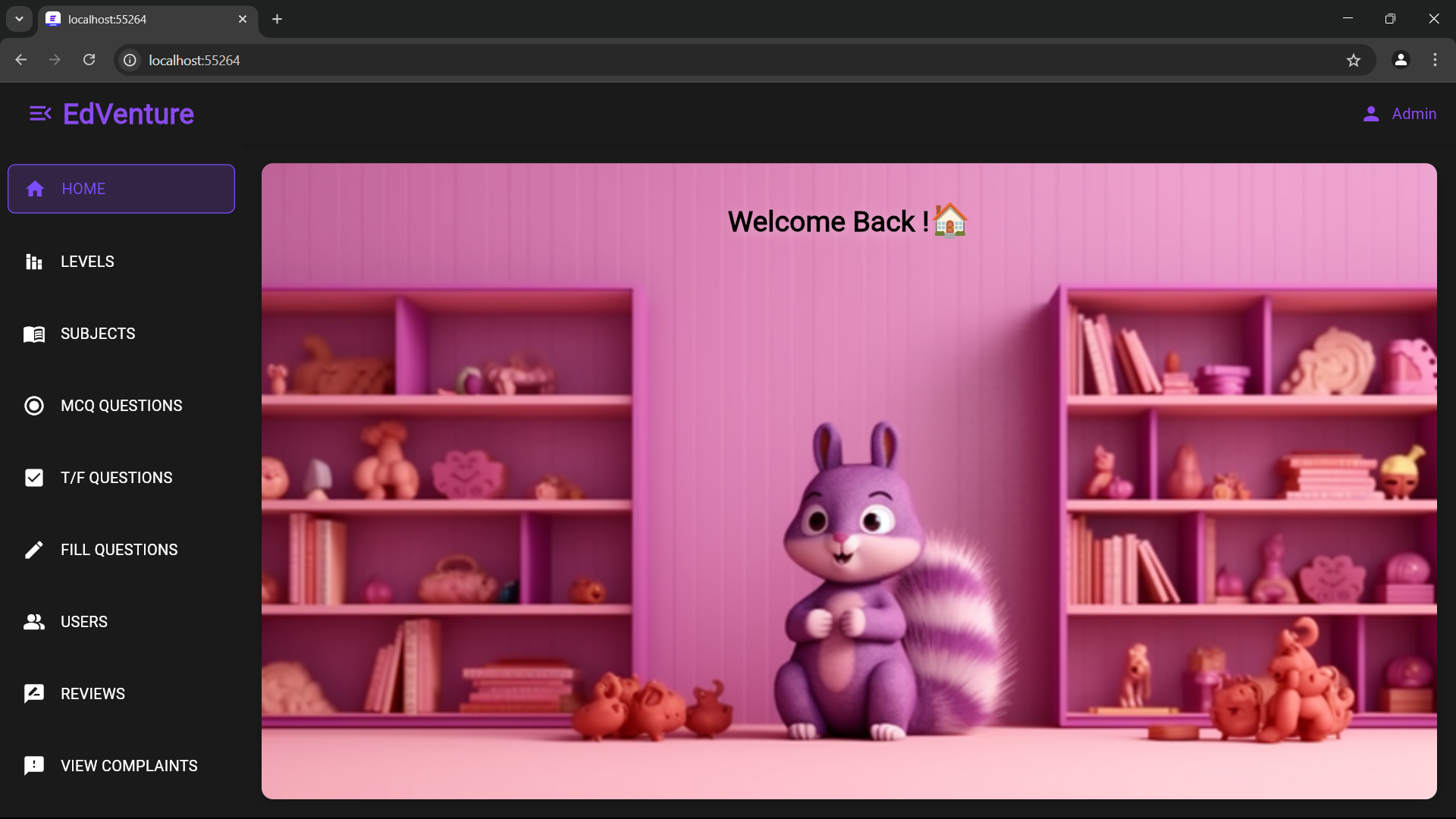
**10. APPENDIX**

**10.1 SCREENSHOTS**

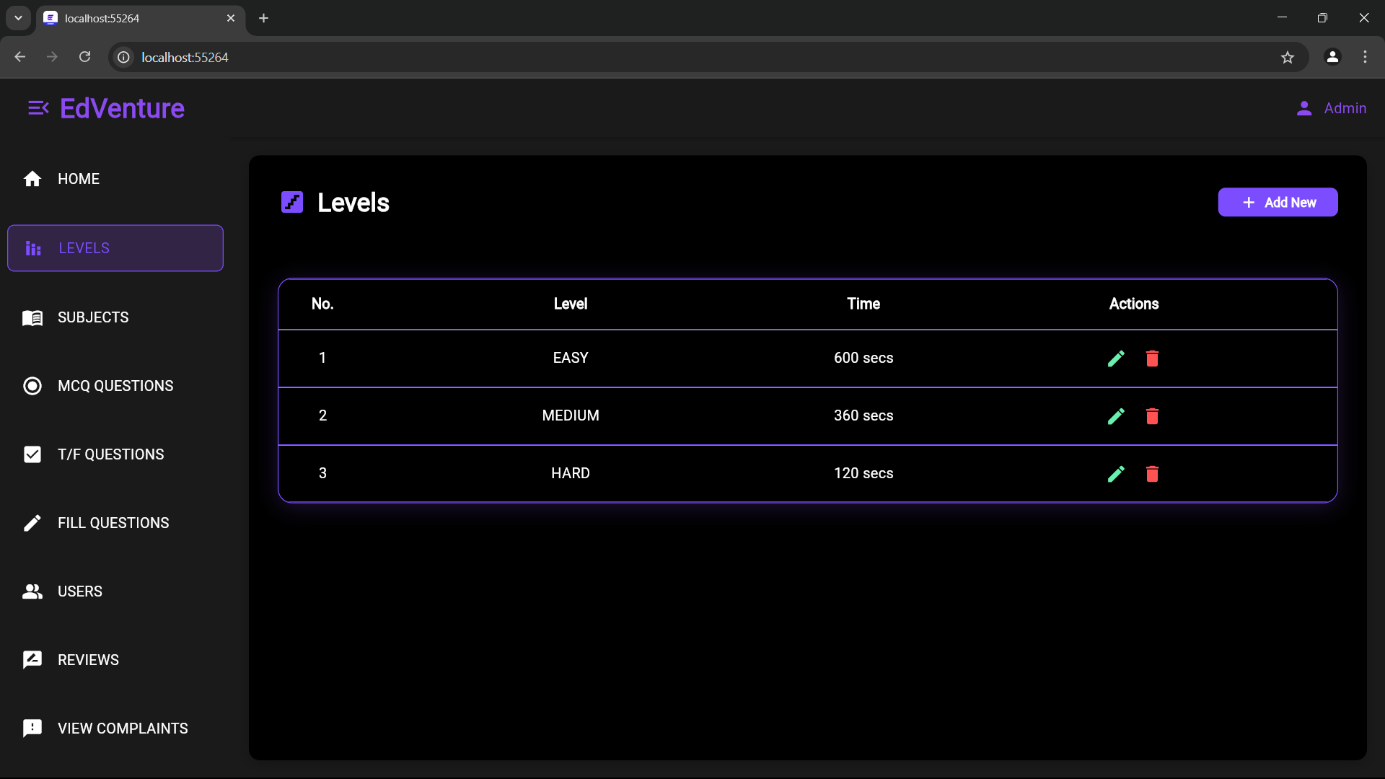
**Admin Login**

****

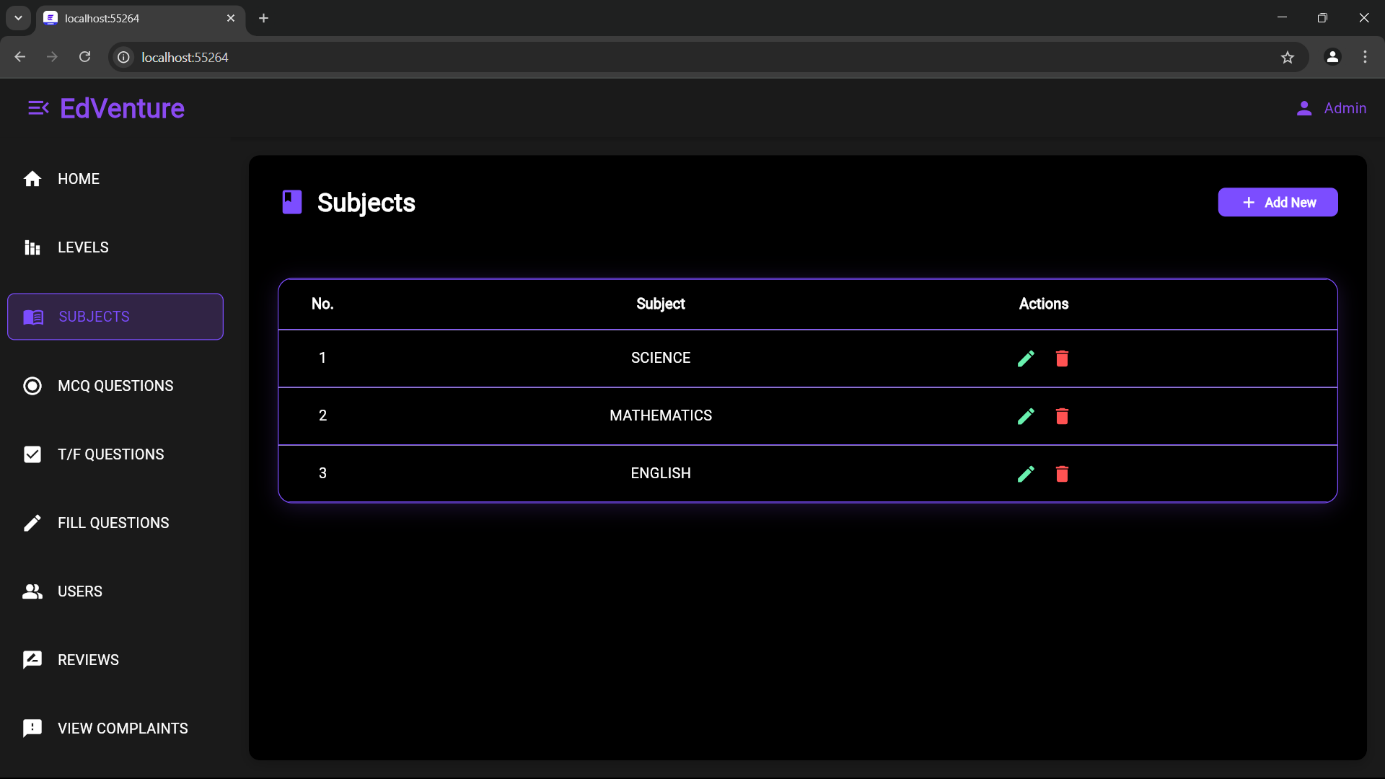
**Admin Dashboard**

****

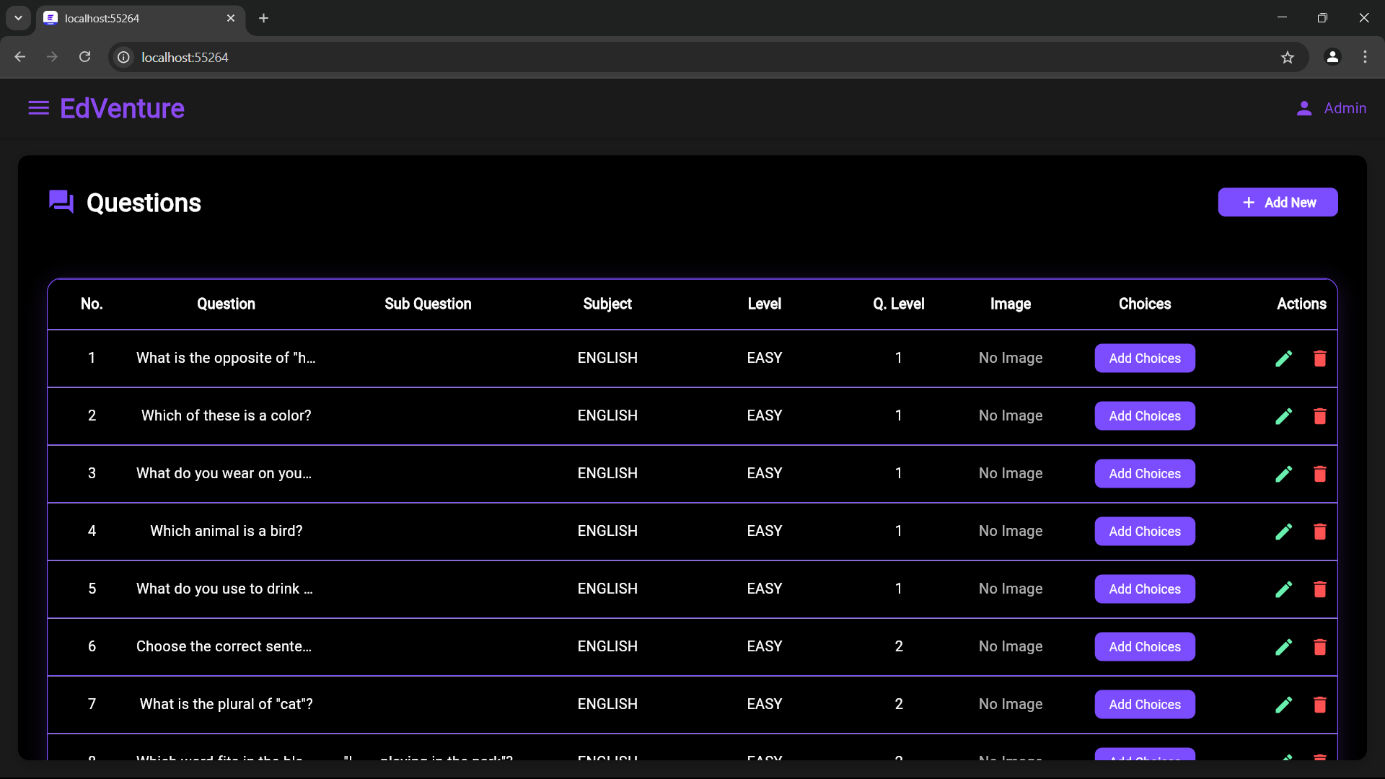
**Level Page**

****

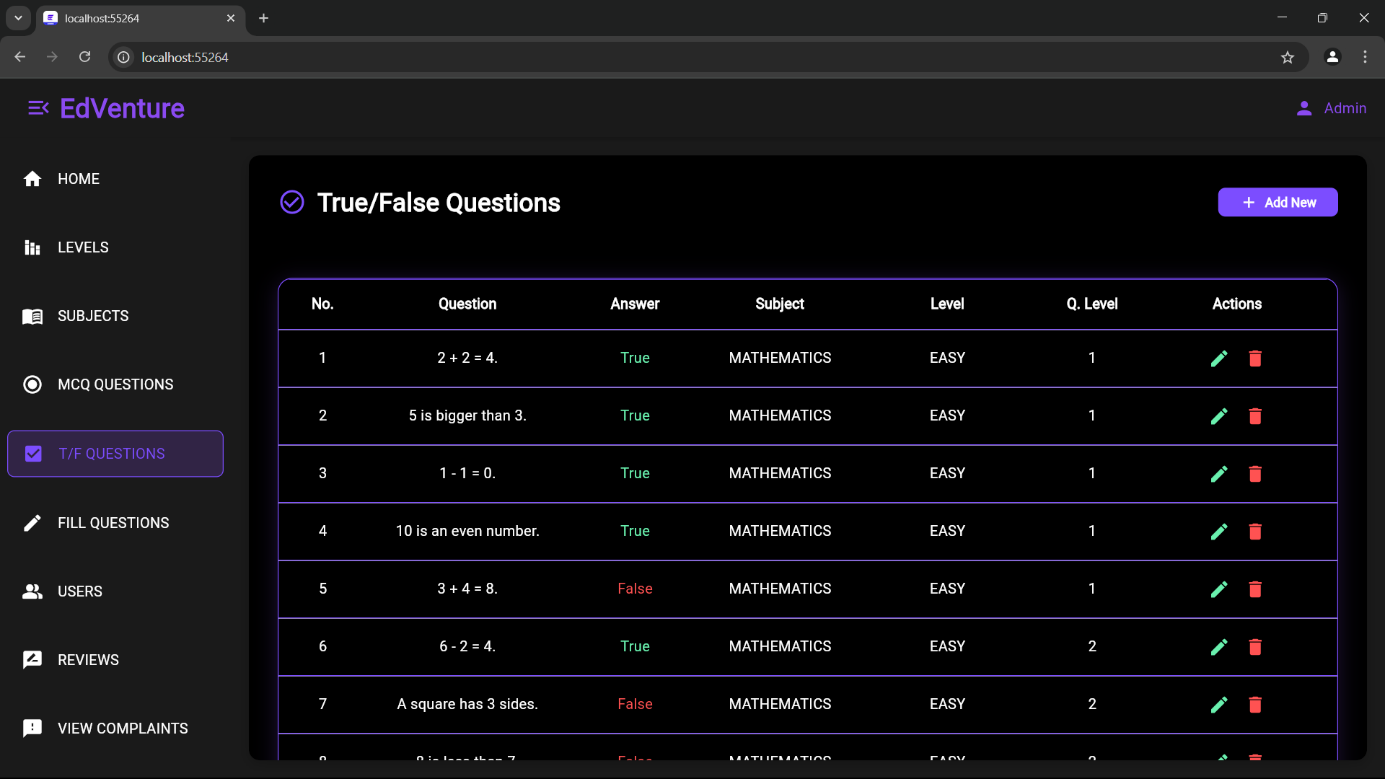
**Subject Page**

****

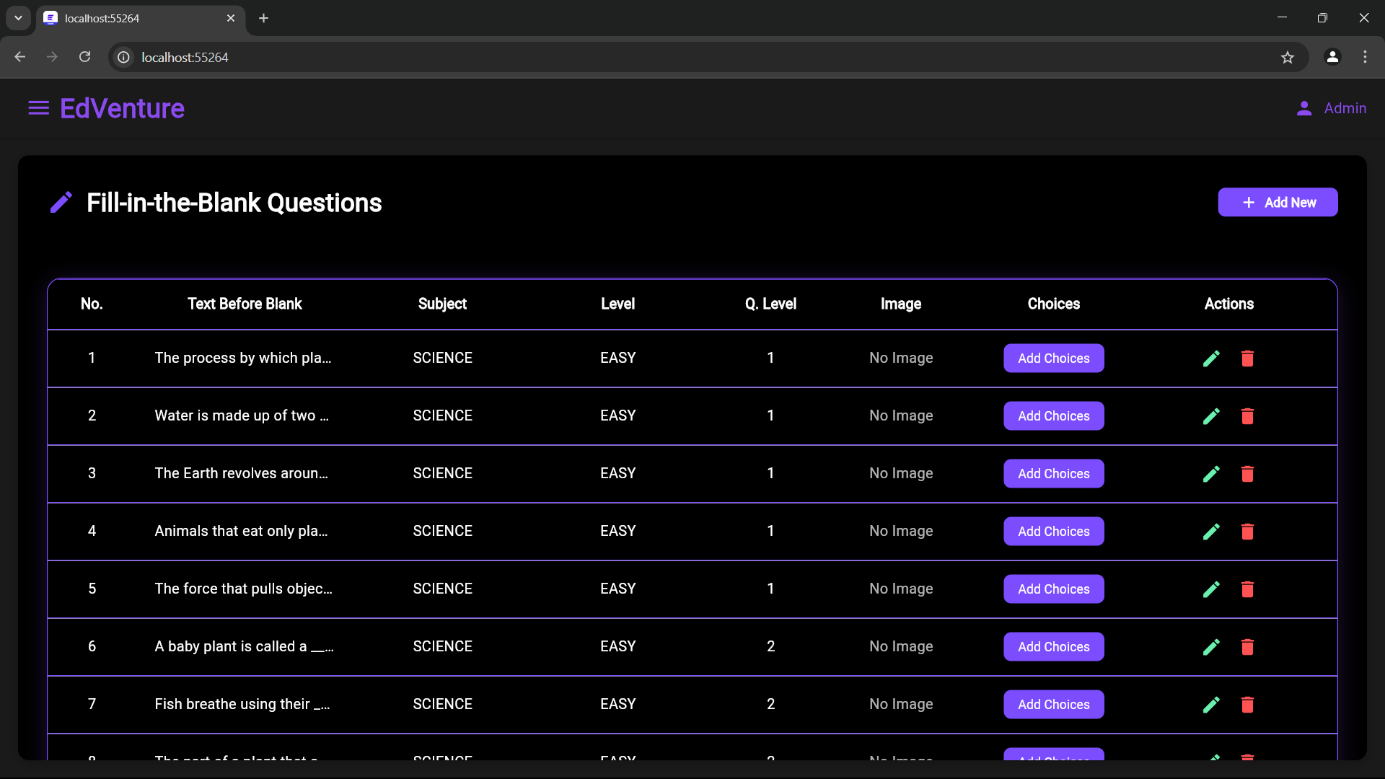
**Mcq Questions**

****

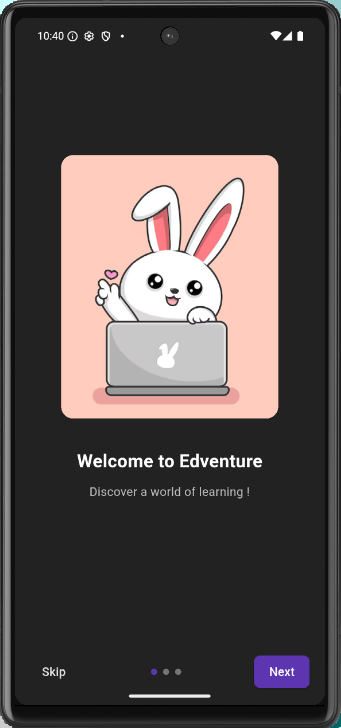
**True or False Questions**

****

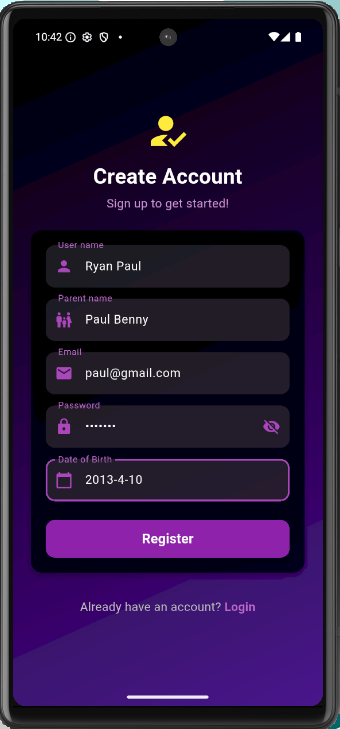
**Fill in the Blanks Questions**

****

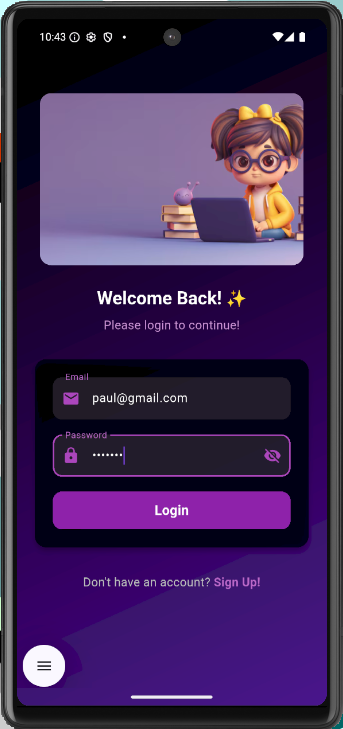
**Intro Page of User**

****

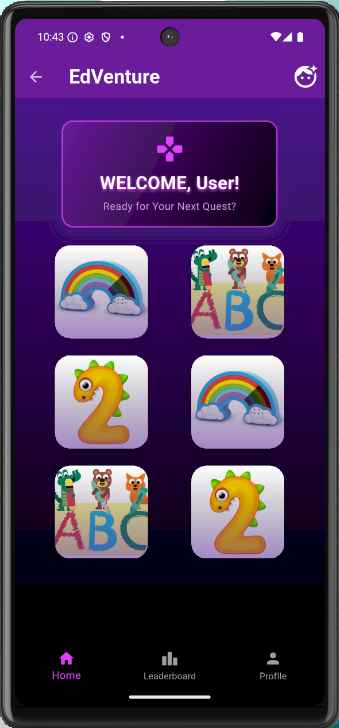
**User Registration**



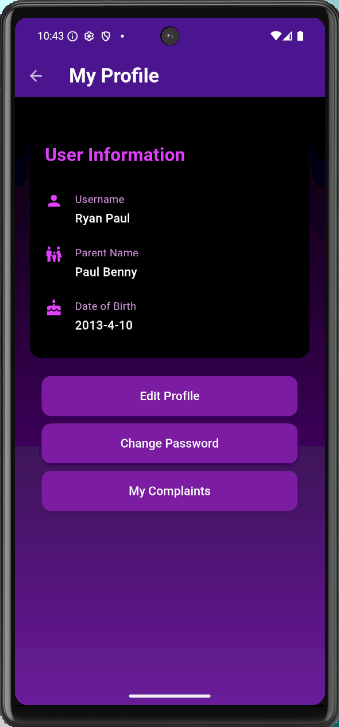
**User Login**

****

**Homepage**



**User Profile**



**10.2 CODE**

**main.dart**

import 'package:flutter/material.dart';

import 'package:supabase\_flutter/supabase\_flutter.dart';

import 'package:user\_edventure/screen/login.dart';

Future<void> main() async {

  await Supabase.initialize(

    url: 'https://zocmpjizmgscrhudkozy.supabase.co',

    anonKey:

        'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6InpvY21waml6bWdzY3JodWRrb3p5Iiwicm9sZSI6ImFub24iLCJpYXQiOjE3MzYzOTk1NzcsImV4cCI6MjA1MTk3NTU3N30.PlwRSf4PIU2DjbnlkzqQqiZ1SfWo5fxCaKCOdT8biqo',

  );

  runApp(MainApp());

}

final supabase = Supabase.instance.client;

class IntroScreen extends StatefulWidget {

  @override

  \_IntroScreenState createState() => \_IntroScreenState();

}

class \_IntroScreenState extends State<IntroScreen> {

  final PageController \_pageController = PageController(initialPage: 0);

  int \_currentPage = 0;

  final int \_totalPages = 3;

  @override

  void dispose() {

    \_pageController.dispose();

    super.dispose();

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      backgroundColor: Theme.of(context).scaffoldBackgroundColor,

      body: Column(

        children: [

          Expanded(

            child: PageView(

              controller: \_pageController,

              onPageChanged: (index) {

                setState(() {

                  \_currentPage = index;

                });

              },

              children: [

                buildPage(

                  imagePath: 'assets/lap.avif',

                  title: 'Welcome to Edventure',

                  subtitle: 'Discover a world of learning !',

                ),

                buildPage(

                  imagePath: 'assets/cube.avif',

                  title: 'Learn Anytime',

                  subtitle: 'Explore learning at your pace !',

                ),

                buildPage(

                  imagePath: 'assets/lap.avif',

                  title: 'Start Your Journey',

                  subtitle: 'Get ready to grow with us !',

                ),

              ],

            ),

          ),

          Padding(

            padding: const EdgeInsets.all(20.0),

            child: Row(

              mainAxisAlignment: MainAxisAlignment.spaceBetween,

              children: [

                TextButton(

                  onPressed: () {

                    Navigator.pushNamed(context, '/login');

                  },

                  child: Text(

                    'Skip',

                    style: TextStyle(

                      color:

                          Theme.of(context).textTheme.labelLarge?.color ??

                          Colors.grey[400],

                      fontSize: 16,

                    ),

                  ),

                ),

                Row(

                  children: List.generate(\_totalPages, (index) {

                    return Container(

                      margin: EdgeInsets.symmetric(horizontal: 4.0),

                      width: 8.0,

                      height: 8.0,

                      decoration: BoxDecoration(

                        shape: BoxShape.circle,

                        color:

                            \_currentPage == index

                                ? Theme.of(context).primaryColor

                                : Colors.grey[600],

                      ),

                    );

                  }),

                ),

                ElevatedButton(

                  onPressed: () {

                    if (\_currentPage < \_totalPages - 1) {

                      \_pageController.nextPage(

                        duration: Duration(milliseconds: 300),

                        curve: Curves.easeInOut,

                      );

                    } else {

                      Navigator.pushNamed(context, '/login');

                    }

                  },

                  style: ElevatedButton.styleFrom(

                    backgroundColor: Theme.of(context).primaryColor,

                    shape: RoundedRectangleBorder(

                      borderRadius: BorderRadius.circular(8),

                    ),

                    padding: EdgeInsets.symmetric(horizontal: 20, vertical: 10),

                  ),

                  child: Text(

                    \_currentPage < \_totalPages - 1 ? 'Next' : 'Get Started',

                    style: TextStyle(

                      color: Theme.of(context).colorScheme.onPrimary,

                      fontSize: 16,

                    ),

                  ),

                ),

              ],

            ),

          ),

        ],

      ),

    );

  }

  Widget buildPage({

    required String imagePath,

    required String title,

    required String subtitle,

  }) {

    return Center(

      child: Column(

        mainAxisAlignment: MainAxisAlignment.center,

        children: [

          ClipRRect(

            borderRadius: BorderRadius.circular(16.0),

            child: Image.asset(

              imagePath,

              width: MediaQuery.of(context).size.width \* 0.7,

              height: 350,

              fit: BoxFit.cover,

              errorBuilder: (context, error, stackTrace) {

                return Container(

                  width: MediaQuery.of(context).size.width \* 0.7,

                  height: 350,

                  color: Colors.grey[800],

                  child: Center(

                    child: Text(

                      'Image not found',

                      style: TextStyle(color: Colors.white),

                    ),

                  ),

                );

              },

            ),

          ),

          SizedBox(height: 40),

          Text(

            title,

            style: TextStyle(

              color: Theme.of(context).colorScheme.onBackground,

              fontSize: 24,

              fontWeight: FontWeight.bold,

            ),

          ),

          SizedBox(height: 12),

          Text(

            subtitle,

            style: TextStyle(

              color: Theme.of(

                context,

              ).colorScheme.onBackground.withOpacity(0.7),

              fontSize: 16,

            ),

            textAlign: TextAlign.center,

          ),

        ],

      ),

    );

  }

}

class MainApp extends StatelessWidget {

  const MainApp({super.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      initialRoute: '/intro',

      theme: ThemeData(

        primarySwatch: Colors.deepPurple,

        primaryColor: Colors.deepPurple[600],

        scaffoldBackgroundColor: Colors.grey[900],

        colorScheme: ColorScheme.fromSwatch(

          primarySwatch: Colors.deepPurple,

          brightness: Brightness.dark,

        ).copyWith(

          secondary: Colors.deepPurple[200],

          onPrimary: Colors.white,

          onBackground: Colors.white,

          onSurface: Colors.white70,

        ),

        textTheme: TextTheme(

          headlineLarge: TextStyle(

            fontSize: 24.0,

            fontWeight: FontWeight.bold,

            color: Colors.white,

          ),

          bodyMedium: TextStyle(fontSize: 16.0, color: Colors.white70),

          labelLarge: TextStyle(fontSize: 16.0, color: Colors.grey[300]),

        ),

        elevatedButtonTheme: ElevatedButtonThemeData(

          style: ElevatedButton.styleFrom(

            backgroundColor: Colors.deepPurple[600],

            shape: RoundedRectangleBorder(

              borderRadius: BorderRadius.circular(8),

            ),

            padding: EdgeInsets.symmetric(horizontal: 20, vertical: 10),

          ),

        ),

      ),

      routes: {

        '/intro': (context) => IntroScreen(),

        '/login': (context) => Login(),

      },

    );

  }

}

**login.dart**

import 'package:flutter/material.dart';

import 'package:supabase\_flutter/supabase\_flutter.dart';

import 'package:user\_edventure/screen/homepg.dart';

import 'package:user\_edventure/screen/register.dart';

class Login extends StatefulWidget {

  const Login({super.key});

  @override

  State<Login> createState() => \_LoginState();

}

class \_LoginState extends State<Login> {

  final TextEditingController \_emailController = TextEditingController();

  final TextEditingController \_passwordController = TextEditingController();

  bool \_isPasswordVisible = false;

  final GlobalKey<FormState> \_formKey = GlobalKey<FormState>();

  final SupabaseClient supabase = Supabase.instance.client;

  Future<void> \_loginUser() async {

    if (!\_formKey.currentState!.validate()) return;

    try {

      final res = await supabase.auth.signInWithPassword(

        email: \_emailController.text.trim(),

        password: \_passwordController.text,

      );

      final user = await supabase

          .from('tbl\_user')

          .select()

          .eq('id', res.user!.id);

      if (user.isNotEmpty) {

        Navigator.pushReplacement(

          context,

          MaterialPageRoute(builder: (context) => HomePage()),

        );

      } else {

        ScaffoldMessenger.of(

          context,

        ).showSnackBar(const SnackBar(content: Text('User not found')));

      }

    } catch (e) {

      ScaffoldMessenger.of(

        context,

      ).showSnackBar(SnackBar(content: Text('Error: $e')));

    }

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      body: Container(

        decoration: BoxDecoration(

          gradient: LinearGradient(

            begin: Alignment.topLeft,

            end: Alignment.bottomRight,

            colors: [Colors.black, Colors.purple[900]!],

          ),

        ),

        child: SingleChildScrollView(

          child: Column(

            mainAxisAlignment: MainAxisAlignment.center,

            crossAxisAlignment: CrossAxisAlignment.center,

            children: [

              const SizedBox(height: 100),

              Padding(

                padding: const EdgeInsets.symmetric(horizontal: 10),

                child: ClipRRect(

                  borderRadius: BorderRadius.circular(15),

                  child: Image.asset(

                    'assets/login.avif',

                    width: MediaQuery.of(context).size.width \* 0.85,

                    height: MediaQuery.of(context).size.height \* 0.25,

                    fit: BoxFit.cover,

                  ),

                ),

              ),

              SizedBox(height: 30),

              RichText(

                text: TextSpan(

                  children: [

                    TextSpan(

                      text: "Welcome Back!",

                      style: TextStyle(

                        fontSize: 24,

                        fontWeight: FontWeight.bold,

                        color: Colors.white,

                      ),

                    ),

                    TextSpan(

                      text: " ✨",

                      style: TextStyle(fontSize: 22, color: Colors.white),

                    ),

                  ],

                ),

              ),

              const SizedBox(height: 10),

              Text(

                "Please login to continue!",

                style: TextStyle(fontSize: 16, color: Colors.purple[200]),

              ),

              SizedBox(height: 30),

              Padding(

                padding: const EdgeInsets.symmetric(horizontal: 20),

                child: Card(

                  shape: RoundedRectangleBorder(

                    borderRadius: BorderRadius.circular(15),

                  ),

                  elevation: 5,

                  color: Colors.black.withOpacity(0.7),

                  child: Padding(

                    padding: const EdgeInsets.all(24),

                    child: Form(

                      key: \_formKey,

                      child: Column(

                        mainAxisSize: MainAxisSize.min,

                        children: [

                          \_buildTextField(

                            controller: \_emailController,

                            label: "Email",

                            icon: Icons.email,

                            keyboardType: TextInputType.emailAddress,

                          ),

                          const SizedBox(height: 20),

                          \_buildPasswordField(),

                          const SizedBox(height: 20),

                          ElevatedButton(

                            style: ElevatedButton.styleFrom(

                              backgroundColor: Colors.purple[600],

                              minimumSize: const Size(double.infinity, 50),

                              shape: RoundedRectangleBorder(

                                borderRadius: BorderRadius.circular(12),

                              ),

                              elevation: 5,

                            ),

                            onPressed: \_loginUser,

                            child: const Text(

                              'Login',

                              style: TextStyle(

                                fontSize: 18,

                                color: Colors.white,

                                fontWeight: FontWeight.bold,

                              ),

                            ),

                          ),

                        ],

                      ),

                    ),

                  ),

                ),

              ),

              const SizedBox(height: 31),

              GestureDetector(

                onTap: () {

                  Navigator.push(

                    context,

                    MaterialPageRoute(builder: (context) => const Register()),

                  );

                },

                child: Padding(

                  padding: const EdgeInsets.only(bottom: 152),

                  child: Row(

                    mainAxisAlignment: MainAxisAlignment.center,

                    children: [

                      Text(

                        "Don't have an account? ",

                        style: TextStyle(color: Colors.grey[400], fontSize: 16),

                      ),

                      Text(

                        "Sign Up!",

                        style: TextStyle(

                          color: Colors.purple[300],

                          fontSize: 16,

                          fontWeight: FontWeight.bold,

                        ),

                      ),

                    ],

                  ),

                ),

              ),

            ],

          ),

        ),

      ),

    );

  }

  Widget \_buildTextField({

    required TextEditingController controller,

    required String label,

    required IconData icon,

    TextInputType keyboardType = TextInputType.text,

  }) {

    return TextFormField(

      controller: controller,

      keyboardType: keyboardType,

      style: const TextStyle(color: Colors.white),

      decoration: InputDecoration(

        labelText: label,

        prefixIcon: Icon(icon, color: Colors.purple[400]),

        filled: true,

        fillColor: Colors.white.withOpacity(0.1),

        border: OutlineInputBorder(

          borderRadius: BorderRadius.circular(12),

          borderSide: BorderSide.none,

        ),

        focusedBorder: OutlineInputBorder(

          borderRadius: BorderRadius.circular(12),

          borderSide: BorderSide(color: Colors.purple[400]!, width: 2),

        ),

        labelStyle: TextStyle(color: Colors.purple[300]),

      ),

      validator: (value) => value!.isEmpty ? "$label cannot be empty" : null,

    );

  }

  Widget \_buildPasswordField() {

    return TextFormField(

      controller: \_passwordController,

      obscureText: !\_isPasswordVisible,

      style: const TextStyle(color: Colors.white),

      decoration: InputDecoration(

        labelText: "Password",

        prefixIcon: Icon(Icons.lock, color: Colors.purple[400]),

        suffixIcon: IconButton(

          icon: Icon(

            \_isPasswordVisible ? Icons.visibility : Icons.visibility\_off,

            color: Colors.purple[400],

          ),

          onPressed: () {

            setState(() {

              \_isPasswordVisible = !\_isPasswordVisible;

            });

          },

        ),

        filled: true,

        fillColor: Colors.white.withOpacity(0.1),

        border: OutlineInputBorder(

          borderRadius: BorderRadius.circular(12),

          borderSide: BorderSide.none,

        ),

        focusedBorder: OutlineInputBorder(

          borderRadius: BorderRadius.circular(12),

          borderSide: BorderSide(color: Colors.purple[400]!, width: 2),

        ),

        labelStyle: TextStyle(color: Colors.purple[300]),

      ),

      validator:

          (value) =>

              value!.length < 6

                  ? "Password must be at least 6 characters"

                  : null,

    );

  }

}