

I. Company Background

History

Antonio Y. Fortich Academy, Incorporated in Bacolod City, has been a cornerstone of early childhood education since its establishment. Over the years, the academy has grown in reputation and size, catering to the educational needs of young children in the community. Despite its success, the academy faced challenges with its manual processes for managing kindergarten classes, such as attendance tracking, grading, and lesson planning. Recognizing the need for modernization, the academy embarked on developing a web application to streamline these processes and enhance overall efficiency.

Description of the Organization

Antonio Y. Fortich Academy is dedicated to providing quality early childhood education with a focus on holistic development. The academy employs a team of dedicated educators and administrative staff who are committed to nurturing young minds. Its mission is to create a supportive and engaging learning environment that fosters intellectual, social, and emotional growth in children. The vision of the academy is to be a leading institution in early childhood education, known for its innovative teaching methods and strong community involvement. The academy's goals include improving operational efficiency, enhancing communication between teachers and parents, and providing a robust educational foundation for its students.

Major Processes and Procedures in the Workplace

The major processes at Antonio Y. Fortich Academy include sales, inventory management, and report generation. The sales process involves enrolling students and managing tuition fees, while inventory management ensures that all educational materials and supplies are adequately stocked and accounted for. Report generation is crucial for tracking student progress, attendance, and academic performance. The new web application will automate these processes, allowing teachers to easily record attendance, update grades, and create lesson plans. This will reduce errors and save time, enabling staff to focus more on teaching and less on administrative tasks. The system will also facilitate better communication with parents through regular updates and notifications.

Products and Services Offered

Antonio Y. Fortich Academy offers a range of educational services designed to cater to the developmental needs of young children. The primary product is its comprehensive kindergarten program, which includes a well-rounded curriculum that covers literacy, numeracy, science, arts, and physical education. The academy also provides additional services such as after-school programs, tutoring, and enrichment activities to support student learning and development. With the implementation of the new web application, the academy will also offer digital tools for parents and teachers to monitor and support student progress, making it easier to stay informed and engaged with their child's education. These offerings aim to provide a supportive and dynamic learning environment that encourages students to achieve their full potential.

II. Entity Relationship Diagram

Entities and Attributes:

Admin

Primary Key: adminID

Attributes: FirstName, LastName, Gender, Age, Street, City, Email, ContactNumber, Password, Image

Attendance

Primary Key: recordID

Foreign Key: studentID

Attributes: Date, AttendanceStatus, Reason

Grade

Primary Key: gradeID

Foreign Key: teacherID, studentID

Attributes: Date, Subject, Grade, Comment

LessonPlan

Primary Key: planID

Foreign Key: teacherID

Attributes: Date, Topic, Resources, Objectives

Student

Primary Key: studentID

Attributes: FirstName, LastName, Gender, Age, Street, City, Email, ContactNumber, Password, ParentDetails, Image

Teacher

Primary Key: teacherID

Attributes: FirstName, LastName, Gender, Age, Street, City, Email, ContactNumber, Password, Image

Relationships

Admin is independent (no foreign keys from other entities).

Attendance relates to Student with a many-to-one relationship (many attendance records for one student).

Grade relates to both Teacher and Student with many-to-one relationships (many grades given by one teacher to one student).

LessonPlan relates to Teacher with a many-to-one relationship (many lesson plans created by one teacher).

Attribute Types

Multi-Valued: tbl_lesson_plan.Resources could be considered as a multi-valued attribute if it's supposed to hold multiple resources.

Derived: No derived attributes are directly shown in the schema.

Composite: Address can be considered as a composite attribute combining Street and City for both tbl_student and tbl_teacher.

III. Normalization Process

3NF Tables already.

tbl_student

studentID INT(11)

◆ FirstName VARCHAR(255)

◆ LastName VARCHAR(255)

◆ Gender VARCHAR(255)

◆ Age VARCHAR(255)

◆ Street VARCHAR(255)

◆ City VARCHAR(255)

◆ Email VARCHAR(255)

◆ ContactNumber VARCHAR(255)

◆ Password VARCHAR(255)

◆ ParentDetails VARCHAR(255)

◆ Image VARCHAR(255)

Indexes

tbl_teacher

teacherID INT(11)

◆ FirstName VARCHAR(255)

◆ LastName VARCHAR(255)

◆ Gender VARCHAR(255)

◆ Age INT(11)

◆ Street VARCHAR(255)

◆ City VARCHAR(255)

◆ Email VARCHAR(255)

◆ ContactNumber VARCHAR(255)

◆ Password VARCHAR(255)

◆ Image VARCHAR(255)

Indexes

tbl_lesson_plan

planID INT(11)

◆ teacherID INT(11)

◆ Date DATE

◆ Topic VARCHAR(255)

◆ Resources VARCHAR(255)

◆ Objectives VARCHAR(255)

Indexes

tbl_admin

adminID INT(11)

◆ FirstName VARCHAR(255)

◆ LastName VARCHAR(255)

◆ Gender VARCHAR(20)

◆ Age INT(11)

◆ Street VARCHAR(255)

◆ City VARCHAR(255)

◆ Email VARCHAR(255)

◆ ContactNumber VARCHAR(255)

◆ Password VARCHAR(255)

◆ Image VARCHAR(255)

Indexes

tbl_attendance

recordID INT(11)

◆ studentID INT(11)

◆ Date DATE

◆ AttendanceStatus VARCHAR(255)

◆ Reason VARCHAR(255)

Indexes

tbl_grade

gradeID INT(11)

◆ Date DATE

◆ teacherID INT(11)

◆ studentID INT(11)

◆ Subject VARCHAR(255)

◆ Grade FLOAT

◆ Comment VARCHAR(255)

Indexes

IV. CRUD Scenarios

Create Scenarios

Scenario 1: A teacher adds a new lesson plan to the portal.

Analysis:

Table: tbl_lesson_plan

Columns: teacherID, Date, Topic, Resources, Objectives

SQL Code:

```
INSERT INTO tbl_lesson_plan (teacherID, Date, Topic, Resources, Objectives)
VALUES (1, '2024-05-20', 'Introduction to Calculus', 'Textbook, Projector', 'Understand
the basics of calculus; Solve simple calculus problems');
```

GUI:

Introduction To Calculus

Date: 2024-05-20

Objectives:

• UNDERSTAND THE BASICS OF CALCULUS; SOLVE SIMPLE CALCULUS PROBLEMS.

Resources:

• TEXTBOOK

• PROJECTOR

Delete

Update

Explanation:

This code inserts a new lesson plan created by a teacher into the tbl_lesson_plan table with the specified details.

Scenario 2: An administrator creates a new user account for a new teacher.

Analysis:

Table: tbl_teacher

Columns: FirstName, LastName, Gender, Age, Street, City, Email, ContactNumber, Password, Image

SQL Code:

```
INSERT INTO tbl_teacher (FirstName, LastName, Gender, Age, Street, City, Email, ContactNumber, Password, Image)
```


```
VALUES ('Jane', 'Doe', 'Female', 30, '123 Maple St', 'Springfield', 'jane.doe@example.com', '+123456789', 'password123', 'jane.jpg');
```

GUI:

Students

Teachers

Add User

	ID Number	First Name	Last Name	Email	Gender	Age	Contact Number	Street	City	Action
	1	Jane	Doe	jane.doe@example.com	female	30	+123456789	123 Maple St	Springfield	<div>Delete</div> <div>Update</div>

Explanation:

This code creates a new teacher account in the `tbl_teacher` table with the specified details.

Read Scenarios

Scenario 1: An admin wants to view the list of students registered in the system.

Analysis:

Table: `tbl_student`

Columns: *

SQL Code:





```
SELECT * FROM tbl_student;
```

GUI:

Teachers

Students

Add User

	ID Number	First Name	Last Name	Email	Gender	Age	Contact Number	Street	City	Action
	1	Emily	Smith	emily@example.com	Female	16	+1987654321	456 Elm St	Los Angeles	<div>Delete</div> <div>Update</div>
	2	Daniel	Wilson	daniel@example.com	male	17	09060880593	789 Maple St	Miami	<div>Delete</div> <div>Update</div>
	11	11	11							<div>Delete</div> <div>Update</div>
	2021202	Erika	Corona							<div>Delete</div> <div>Update</div>

Explanation:

This code retrieves all student records from the tbl_student table for the admin to view.

Scenario 2: A teacher wants to view their lesson plans.

Analysis:

Table: tbl_lesson_plan

Columns: *

Condition: teacherID = ?

SQL Code:

```
SELECT * FROM tbl_lesson_plan WHERE teacherID = 1;
```

GUI:

Lesson Plan

Add Lesson Plan

Introduction To Calculus		Date: 2024-05-20
Objectives:	Resources:	
<ul style="list-style-type: none">UNDERSTAND THE BASICS OF CALCULUS; SOLVE SIMPLE CALCULUS PROBLEMS.	<ul style="list-style-type: none">TEXTBOOKPROJECTOR	
<div>DeleteUpdate</div>		

Introduction To Algebra		Date: 2024-12-31
Objectives:	Resources:	
<ul style="list-style-type: none">QWEQEWQ	<ul style="list-style-type: none">EWQEWEQWEQWEQ	

Explanation:

This code retrieves lesson plans created by a specific teacher.

Update Scenarios

Scenario 1: A teacher updates the topic of a previously created lesson plan.

Analysis:

Table: tbl_lesson_plan

Columns: Topic

Condition: planID = ?

SQL Code:

```
UPDATE tbl_lesson_plan
SET Topic = 'Advanced Calculus'
WHERE planID = 7;
```

GUI:

Update Lesson Plan

Date

20/05/2024

Topic

Advanced Calculus

Resources

Textbook

Projector

Explanation:

This code updates the topic of a specific lesson plan in the `tbl_lesson_plan` table.

Scenario 2: An administrator updates a student's contact information.

Analysis:

Table: `tbl_student`

Columns: `ContactNumber`

Condition: `studentID = ?`

SQL Code:

```
UPDATE tbl_student
SET ContactNumber = '+123987654'
WHERE studentID = 1;
```

GUI:

First Name	Emma
Last Name	Clark
Email	
Age	35
Street	567 Birch St
City	Houston
Gender	Male
Contacts	+123987654

Update

Close

Explanation:

This code updates the contact information of a specific student in the `tbl_student` table.

Delete Scenarios

Scenario 1: A teacher deletes a lesson plan that is no longer relevant.

Analysis:

Table: `tbl_lesson_plan`

Columns: `planID`

SQL Code:

```
DELETE FROM tbl_lesson_plan
WHERE planID = 9;
```

GUI:

Lesson Plan

Add Lesson Plan

Introduction To Algebra

Objectives:

- QWEQEWQ

Resources:

- EWQEWEQWEQWEQ

Delete

Update

Date: 2024-12-31

Explanation:

This code deletes a specific lesson plan from the `tbl_lesson_plan` table.

Scenario 2: An administrator removes a student who has graduated from the school.

Analysis:

Table: `tbl_student`

Columns: `studentID`

SQL Code:





```
DELETE FROM tbl_student
WHERE studentID = 2;
```

GUI:

Teachers

Students




Add User

	ID Number	First Name	Last Name	Email	Gender	Age	Contact Number	Street	City	Action
	1	Emily	Smith	emily@example.com	Female	16	+1987654321	456 Elm St	Los Angeles	<div>Delete</div> <div>Update</div>
	2	Daniel	Wilson	daniel@example.com	male	17	09060880593	789 Maple St	Miami	<div>Delete</div> <div>Update</div>
	11	11	11							<div>Delete</div> <div>Update</div>
	2021202	Erika	Corona							<div>Delete</div> <div>Update</div>

Teachers

Students

Add User

	ID Number	First Name	Last Name	Email	Gender	Age	Contact Number	Street	City	Action
	1	Emily	Smith	emily@example.com	Female	16	+1987654321	456 Elm St	Los Angeles	<div>Delete</div> <div>Update</div>
	2	Daniel	Wilson	daniel@example.com	male	17	09060880593	789 Maple St	Miami	<div>Delete</div> <div>Update</div>
	2021202	Erika	Corona							<div>Delete</div> <div>Update</div>

A list or table with student records and a delete button for each entry.

Explanation:

This code deletes a specific student from the tbl_student table.

Scenario 3: An admin removes a teacher who has left the school.

Analysis:

Table: tbl_teacher

Columns: teacherID

SQL Code:





```
DELETE FROM tbl_teacher
WHERE teacherID = 3;
```

GUI:

Students

Teachers

Add User

	ID Number	First Name	Last Name	Email	Gender	Age	Contact Number	Street	City	Action
	1	Jane	Doe	jane.doe@example.com	female	30	+123456789	123 Maple St	Springfield	<div>Delete</div> <div>Update</div>
	2	Emma	Clark		male	35	+123987654	567 Birch St	Houston	<div>Delete</div> <div>Update</div>
	3	Sophia	Thomas		Female	45	+1654321890	678 Elm St	Atlanta	<div>Delete</div> <div>Update</div>
	10	Julie	Gerogalin	lovikim1@gmail.com	female	20	0907088049	Lacson	Himamaylan	<div>Delete</div> <div>Update</div>

A list or table with teacher records and a delete button for each entry.

Explanation:

This code deletes a specific teacher from the tbl_teacher table.