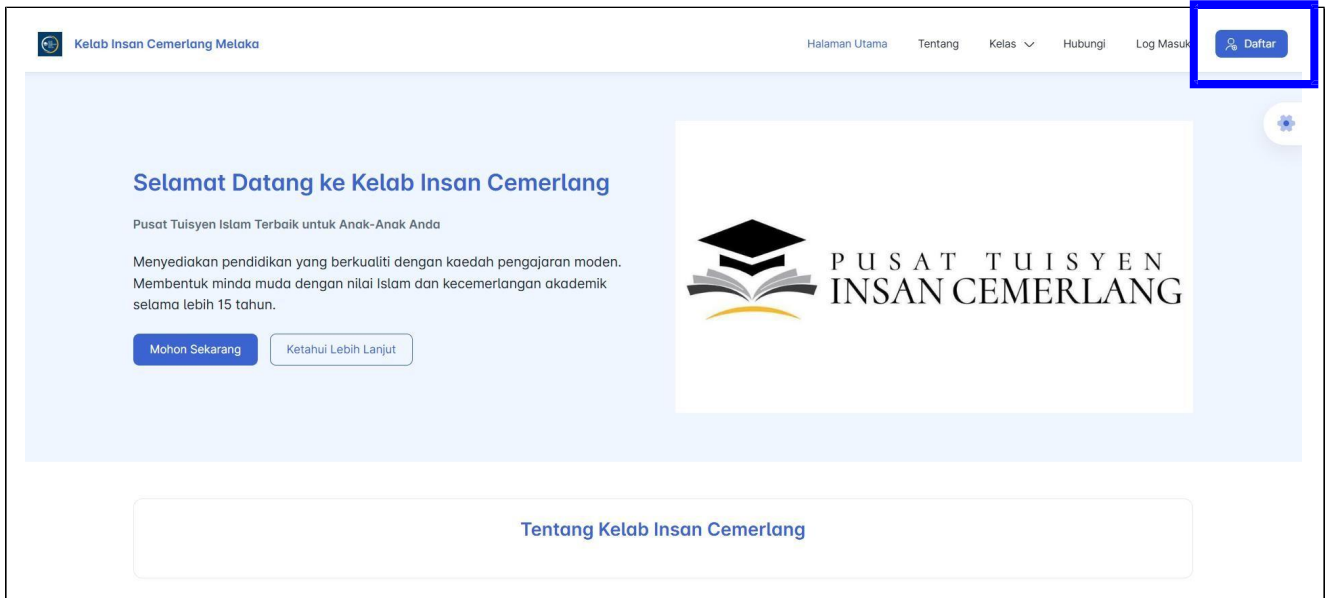
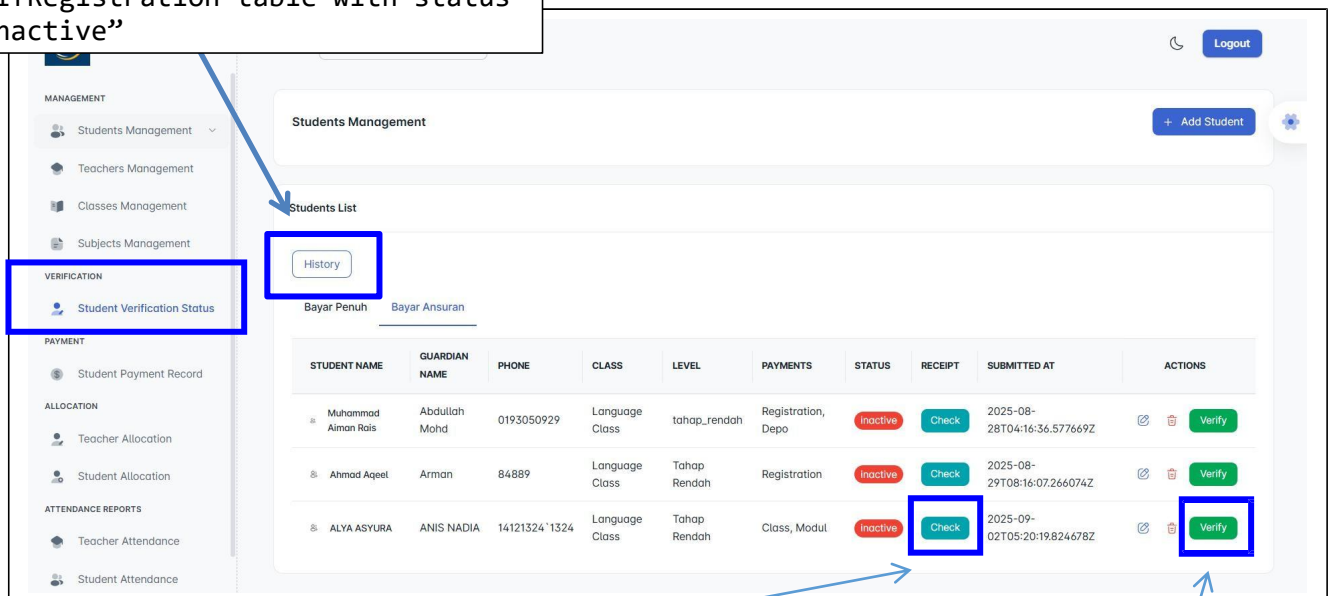


Client click here to do the SelfRegistration



Admin can view which student have verified before *which I took from SelfRegistration table with status "inactive"

After client registered . It will go to "Student Verification Status" - Admin Dashboard



Admin can view receipt that uploaded from client . (Self-Registration)
They need to make sure the amount from the receipt and also make sure if there's transactions from client or not.

Admin can verify if they confirmed there's transaction from client and received payment .

*Verify means change the status to "Active".

it will automatically create and save in :
Student Table and Payment Table

Students Management

+ Add Student

Students List

History

Bayar Penuh Bayar Ansur

Verify Student

Are you sure you want to verify ALYA ASYURA? This will set their status to active.

Cancel Confirm

STUDENT NAME	GUARDIAN NAME	LEVEL	CLASS	STATUS	RECEIPT	SUBMITTED AT	ACTIONS		
Muhammad Almar Rais	Abdullah Mohd	0193050929	Language Class	tahap_rendah	Registration, Depo	inactive	Check	2025-08-28T04:16:36.577669Z	Verify
Ahmad Aqeel	Arman	84889	Language Class	Tahap Rendah	Registration	inactive	Check	2025-08-29T08:16:07.266074Z	Verify
ALYA ASYURA	ANIS NADIA	14121324`1324	Language Class	Tahap Rendah	Class, Modul	inactive	Check	2025-09-02T05:20:19.824678Z	Verify



After Admin confirm to verify :
It will automatically save
in : "Student" and "Payment"

MANAGEMENT

Students Management

Student Details

Student Class Allocation

Teachers Management

Classes Management

Subjects Management

VERIFICATION

Student Verification Status

PAYMENT

Student Payment Record

ALLOCATION

Teacher Allocation

Student Allocation

ATTENDANCE REPORTS

Students Management

+ Add Student

Students List

STUDENT NAME	GUARDIAN NAME	LEVEL	STATUS	ENROLLMENT DATE	ACTIONS
Ahmad Aqeel	Ali Bin Abu	Primary	active	2025-08-26	
Aqeel Ammar	Zulfqar	Primary	active	2025-08-26	
Ayra Aryssa	Aminur Rasyid	Tahap Rendah	active	2025-09-02	
Ahmad Ridzuan	Rohaya Hatta	Tahap Rendah	active	2025-09-02	
Abdullah Mohd	Pasraisyr Ready	Tahap Rendah	active	2025-09-02	
ALYA ASYURA	ANIS NADIA	Tahap Rendah	active	2025-09-02	

MANAGEMENT

Students Management

Teachers Management

Classes Management

Subjects Management

VERIFICATION

Student Verification Status

PAYMENT

Student Payment Record

ALLOCATION

Payment Record

Manage payment records for students

+ Add Payment Record

Payment Records

STUDENT NAME	STUDENT IC	STUDENT CLASS	PAYMENT METHOD	PAYMENT REFERENCE	RECEIPT	SELECTED PAYMENTS	TOTAL PAYMENT	SUBMITTED AT	ACTIONS
Abdullah Mohd	433344343133	Language Class	bank_transfer	Ahmad	View	registration: 20 depo: 50	70.00	2025-09-02T10:17:26.319045Z	
ALYA ASYURA	13123123123	Language Class	bank_transfer	Ahmad	View	class modul	0.00	2025-09-02T10:54:55.952132Z	

My backend models looks like : from SelfRegistration to Student and Payment

```
class SelfRegistration(models.Model):
    guardianName = models.CharField(max_length=100)
    guardianIC = models.CharField(max_length=20)
    guardianPhone = models.CharField(max_length=20)
    studentName = models.CharField(max_length=100)
    studentIC = models.CharField(max_length=20)
    address = models.TextField()
    studentClass = models.CharField(max_length=50)
    level = models.CharField(max_length=50)
    class_method = models.CharField(max_length=50)
    payment_method = models.CharField(max_length=50)
    payment_reference = models.CharField(max_length=100)
    payment_receipt = models.FileField(upload_to='receipts/')
    selected_payments = models.JSONField(default=list) # Stores selected payment items
    total_payment = models.DecimalField(max_digits=10, decimal_places=2) # Stores total
payment amount
    submitted_at = models.DateTimeField(auto_now_add=True)
    STATUS_CHOICES = [
        ('active', 'Active'),
        ('inactive', 'Inactive'),
    ]
    status = models.CharField(max_length=8, choices=STATUS_CHOICES, default='inactive') #
Staff sets this

    def save(self, *args, **kwargs):
        was_inactive = False
        if self.pk:
            old = SelfRegistration.objects.get(pk=self.pk)
            was_inactive = old.status == 'inactive'
            super().save(*args, **kwargs)
            # If status changed to active, promote to Student
        if self.status == 'active' and (not self.pk or was_inactive):
            from ttp1_backend.models import Student, Class, Payment
            try:
                class_obj = Class.objects.get(id=self.studentClass)
            except Class.DoesNotExist:
                return
            Student.objects.get_or_create(
                studentIC=self.studentIC,
                defaults={
                    'guardianName': self.guardianName,
                    'guardianIC': self.guardianIC,
                    'guardianPhone': self.guardianPhone,
                    'studentName': self.studentName,
                    'address': self.address,
                    'level': self.level,
                    'status': 'active',
                    'enrollmentDate': self.submitted_at.date()
                }
            )
            Payment.objects.create( studentName=self.s
                tudentName, studentIC=self.studentIC,
                studentClass=class_obj.name if class_obj else None,
                payment_method=self.payment_method,
                payment_reference=self.payment_reference,
                payment_receipt=self.payment_receipt,
                selected_payments=self.selected_payments,
                total_payment=self.total_payment,
                submitted_at=self.submitted_at
            )
```

*I'm not an expert, but this is how the flow looks like.

*The admin can see which clients registered by themselves through the website.

Class Payment

```
class Payment(models.Model):
    studentName = models.CharField(max_length=100)
    studentIC = models.CharField(max_length=20)
    studentClass = models.CharField(max_length=50, blank=True, null=True)
    payment_method = models.CharField(max_length=50)
    payment_reference = models.CharField(max_length=100)
    payment_receipt = models.FileField(upload_to='receipts/')
    selected_payments = models.JSONField(default=list)
    total_payment = models.DecimalField(max_digits=10, decimal_places=2,
default=0.00)
    submitted_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f"Payment {self.payment_reference} for {self.studentIC}"
```

Class Student

```
class Student(models.Model):
    STATUS_CHOICES = [
        ('active', 'Active'),
        ('inactive', 'Inactive'),
    ]
    guardianName = models.CharField(max_length=255)
    guardianIC = models.CharField(max_length=32)
    guardianPhone = models.CharField(max_length=32)
    studentName = models.CharField(max_length=255)
    studentIC = models.CharField(max_length=32)
    address = models.TextField()
    level = models.CharField(max_length=64)
    status = models.CharField(max_length=8, choices=STATUS_CHOICES,
default='active')
    enrollmentDate = models.DateField()

    def __str__(self):
        return self.studentName
```

Admin can add Payment by student for Monthly payment .

*Parents and Admin will contact by ws for montly payment. This is just for Admin to record and save in system.

*later on will make tabs that class them by "Payment Registration" and "Monthly"

1. Payment Registration including : Depo , Modul , Registration Fee , Class
2. Monthly : (example : CLASS KAFA RM 50 , PAY RM 50 EVERYMONTH)

Payment Record

Manage payment records for students

Add Payment Record

STUDENT NAME	STUDENT IC	STUDENT CLASS	PAYMENT METHOD	PAYMENT REFERENCE	RECEIPT	SELECTED PAYMENTS	TOTAL PAYMENT	SUBMITTED AT	ACTIONS
Abdullah Mohd	433344343133	Language Class	bank_transfer	Ahmad	View	registration: 20 depo: 50	70.00	2025-09-02T10:17:26.319045Z	Edit Delete
ALYA ASYURA	13123123123	Language Class	bank_transfer	Ahmad	View	class modul	0.00	2025-09-02T10:54:55.952132Z	Edit Delete

Add Payment Record

Student Name
Ahmad Aqeel

Student IC
0104282163

Student Class

Payment Method

Payment Reference

Total Payment
0

Upload Receipt

Month: January Amount: 50 Remove

Month: February Amount: 50 Remove

Add Payment

Cancel Add Payment Record

Fetch from Student

Auto-fill from selected student

JSON format that allows adding multiple months if a client wants to pay in advance for the upcoming months

MANAGEMENT

Students Management

Teachers Management

Classes Management

Subjects Management

VERIFICATION

Student Verification Status

PAYMENT

Student Payment Record

ALLOCATION

Teacher Allocation

Student Allocation

ATTENDANCE REPORTS

Teacher Attendance

Home > Admin ICIM

Admin ICIM

Class Management

Manage class details

Classes List

CLASS NAME

Kelas Mer

Language

Program

Class Name

Subjects

English, Python

English

Arabic

Python

Memorah

Start Time

End Time

Price

0

Status

Active

Cancel

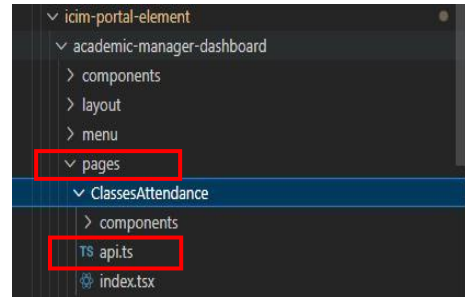
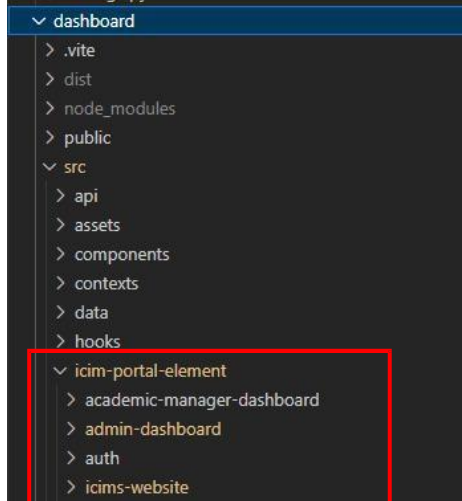
Add Class

+ Add Class

PRICE	STATUS	ACTIONS
250		
250		

Subjects field can be multiple .
One class have many Subjects

Project ICIM directory :



*each pages have the api.ts itself

```
# Dashboard :  
academic-manager-dashboard (same function with Admin)  
admin-dashboard  
teacher-dashboard  
  
Frontend Folder :  
icims-website
```

```
# every ["pages" folder] have their own parents and api.ts
```

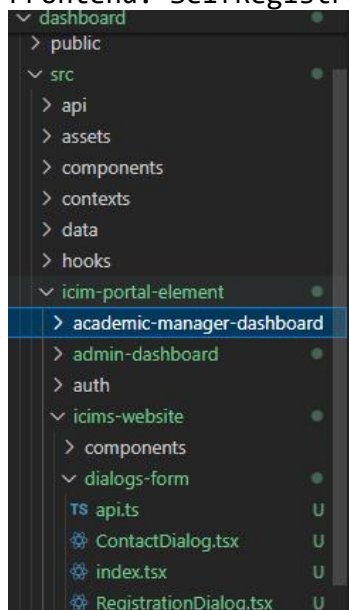
example parents :

```
icim-portal-element > academic-manager-dashboard > pages > ClassesManagement > index.tsx
```

example api.ts:

```
icim-portal-element > academic-manager-dashboard > pages > ClassesManagement > api.ts
```

Frontend: SelfRegistration form



```
Icim-portal-element > icims-website>  
dialogs-form> api.ts
```