

Welcome Back!

Login in to access your hostel dashboard

Log In Enter your credentials to continue Username (Roll or Email) **Password** Role Admin Login



All Students

Name	Roll No	Room	Last Active	IP Address	Actions
Alice Smith	101	A-101	N/A	N/A	Change Room Edit Delete
Bob Johnson	102	Unassigned	N/A	N/A	Assign Room Edit Delete
Mahak Varshney	104	A-101	2025-10-05 20:51:33 (Inactive)	127.0.0.1	Change Room Edit Delete

Add Student

Staff Members

Name	Department	Email	Actions
Jane Doe	Administration	jane.doe@example.com	Edit Delete
suraj	Maintenance	s12@gmail.com	Edit Delete

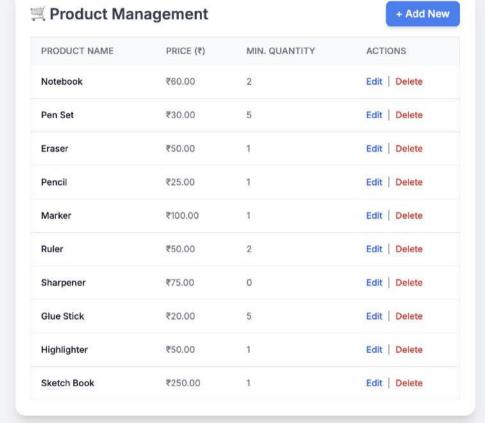
Add Staff

Rooms

Room No	Capacity	Occupied	Available
A-101	2	2	0
A-102	3	0	3
A-103	2	0	2
A-104	2	0	2
A-105	3	0	.3
B-101	2	0	2
B-102	2	0	2
B-103	3	0	3

Staff Dashboard

Logout



Orders (0)

No orders found.

Pending Gate Passes (0)

No pending gate pass requests.

Rooms

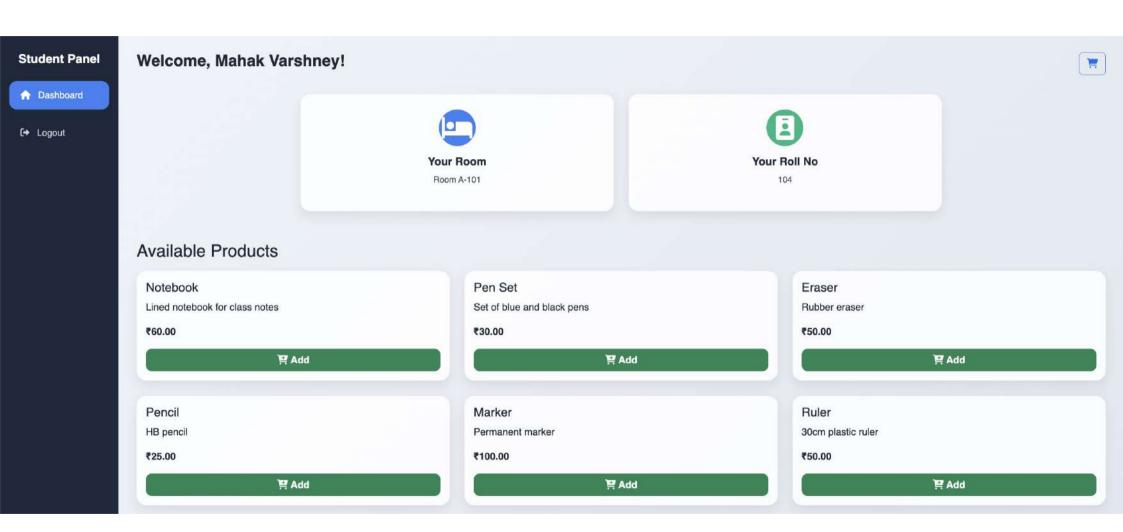
Room No	Capacity	Occupied	Available
A-101	2	2	0
A-102	3	0	3
A-103	2	0	2
A-104	2	0	2
A-105	3	0	3
B-101	2	0	2
B-102	2	0	2
B-103	3	0	3
B-104	2	0	2
B-105	3	0	3
C-101	2	0	2
C-102	2	0	2

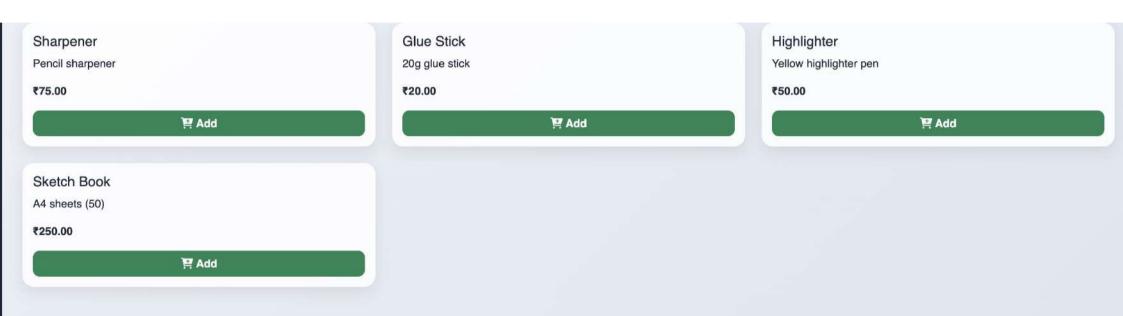
Student Complaints

Student	Title	Description	Status	Action	
The State of the S	The state of the s	The state of the s	District the state of the state		

■ Gate Pass Requests

Student	Reason	Status	Requested At		
No gate pass requests found.					





You have no gate pass requests.

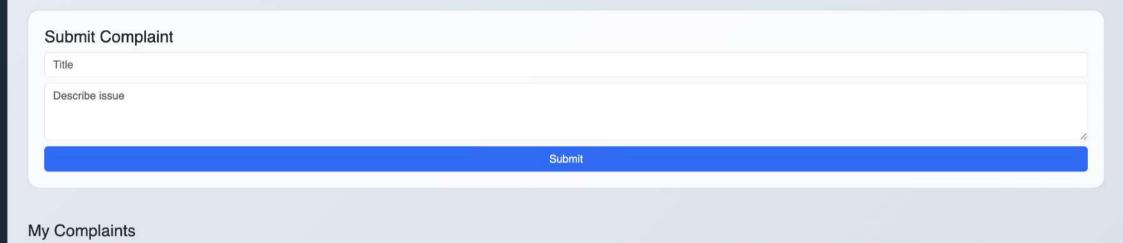
Request New Gate Pass

Your Gate Passes

Fees Summary

Month	Due	Paid	Balance	Status
October 2025	₹10000.00	₹10000.00	₹0.00	Paid
November 2025	₹10000.00	₹0.00	₹10000.00	Not required now
December 2025	₹10000.00	₹0.00	₹10000.00	Not required now
January 2026	₹10000.00	₹0.00	₹10000.00	Not required now
February 2026	₹10000.00	₹0.00	₹10000.00	Not required now
March 2026	₹10000.00	₹0.00	₹10000.00	Not required now
April 2026	₹10000.00	₹0.00	₹10000.00	Not required now
May 2026	₹10000.00	₹0.00	₹10000.00	Not required now
June 2026	₹10000.00	₹0.00	₹10000.00	Not required now
July 2026	₹10000.00	₹0.00	₹10000.00	Not required now
August 2026	₹10000.00	₹0.00	₹10000.00	Not required now
September 2026	₹10000.00	₹0.00	₹10000.00	Not required now

Pay Fees =



No complaints submitted yet.

Submitted On

Status

Title

Description

```
from extensions import db
from werkzeug.security import generate_password_hash, check_password_hash
from datetime import datetime
# All classes will inherit from db.Model
class Admin(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True, nullable=False)
    password_hash = db.Column(db.String(128))
    def set_password(self, password):
        self.password_hash = generate_password_hash(password)
    def check password(self, password):
        return check_password_hash(self.password_hash, password)
class Staff(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    email = db.Column(db.String(120), unique=True, nullable=False)
    department = db.Column(db.String(100))
    password_hash = db.Column(db.String(128))
    def set_password(self, password):
        self.password_hash = generate_password_hash(password)
    def check_password(self, password):
        return check_password_hash(self.password_hash, password)
class Room(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    room_no = db.Column(db.String(10), unique=True, nullable=False)
    capacity = db.Column(db.Integer, nullable=False)
    occupied = db.Column(db.Integer, default=0)
    students = db.relationship('Student', back_populates='room') # link to students
class Student(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    roll = db.Column(db.String(20), unique=True, nullable=False)
    name = db.Column(db.String(100), nullable=False)
    email = db.Column(db.String(100), nullable=False)
    password_hash = db.Column(db.String(128))
    room_id = db.Column(db.Integer, db.ForeignKey('room.id')) # FK to Room
    room = db.relationship('Room', back_populates='students') # back-reference
    last_active = db.Column(db.DateTime, nullable=True)
    last_active_ip = db.Column(db.String(45), nullable=True)
    is_assigned = db.Column(db.Boolean, default=False)
    def set_password(self, password):
        self.password_hash = generate_password_hash(password)
    def check_password(self, password):
        return check_password_hash(self.password_hash, password)
```

```
class Product(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    description = db.Column(db.String(255))
    price = db.Column(db.Float, nullable=False)
    stock = db.Column(db.Integer, nullable=False)
    min_quantity = db.Column(db.Integer, default=1)
class Complaint(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    student_id = db.Column(db.Integer, db.ForeignKey('student.id'))
    title = db.Column(db.String(100))
    description = db.Column(db.Text)
    status = db.Column(db.String(50), default='Pending')
    date_submitted = db.Column(db.DateTime, default=datetime.utcnow)
    student = db.relationship("Student", backref="complaints")
class Cart(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    student_id = db.Column(db.Integer, db.ForeignKey('student.id'), nullable=False)
    product id = db.Column(db.Integer, db.ForeignKey('product.id'), nullable=False)
    quantity = db.Column(db.Integer, nullable=False)
    student = db.relationship('Student', backref=db.backref('cart_items', lazy=True))
    product = db.relationship('Product', backref=db.backref('cart_items', lazy=True))
class GatePass(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    student_id = db.Column(db.Integer, db.ForeignKey('student.id'))
    reason = db.Column(db.String(200), nullable=False)
    leave_date = db.Column(db.Date, nullable=False)
    return_date = db.Column(db.Date, nullable=False)
    status = db.Column(db.String(20), default='Pending')
    date requested = db.Column(db.DateTime, default=datetime.utcnow)
    student = db.relationship('Student', backref='gate_passes')
class Fees(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    student_id = db.Column(db.Integer, db.ForeignKey('student.id'), nullable=False)
    month = db.Column(db.String(20), nullable=False)
    year = db.Column(db.Integer, nullable=False)
    amount_due = db.Column(db.Float, default=0)
    amount_paid = db.Column(db.Float, default=0)
    status = db.Column(db.String(20), default='Pending')
    student = db.relationship('Student', backref='fees')
    @property
    def balance(self):
        return self.amount_due - self.amount_paid
```

```
114
      class Order(db.Model):
115
          id = db.Column(db.Integer, primary_key=True)
116
          student_id = db.Column(db.Integer, db.ForeignKey('student.id'), nullable=False)
117
          product_id = db.Column(db.Integer, db.ForeignKey('product.id'), nullable=False)
118
          quantity = db.Column(db.Integer, nullable=False)
119
          total price = db.Column(db.Float, nullable=False)
120
          status = db.Column(db.String(50), default="Pending")
121
          ordered at = db.Column(db.DateTime, default=datetime.utcnow)
122
          student = db.relationship('Student', backref='orders')
123
124
          product = db.relationship('Product', backref='orders')
```