



Welcome Back!

Login in to access your hostel dashboard

Log In

Enter your credentials to continue

Username (Roll or Email)

Password

Role

Admin

▼

Login

Welcome, Admin 🏠💻

📋 All Students

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

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

Product Management

+ Add New

PRODUCT NAME	PRICE (₹)	MIN. QUANTITY	ACTIONS
Notebook	₹60.00	2	Edit Delete
Pen Set	₹30.00	5	Edit Delete
Eraser	₹50.00	1	Edit Delete
Pencil	₹25.00	1	Edit Delete
Marker	₹100.00	1	Edit Delete
Ruler	₹50.00	2	Edit Delete
Sharpener	₹75.00	0	Edit Delete
Glue Stick	₹20.00	5	Edit Delete
Highlighter	₹50.00	1	Edit Delete
Sketch Book	₹250.00	1	Edit Delete

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
---------	-------	-------------	--------	--------

Gate Pass Requests

Student	Reason	Status	Requested At
---------	--------	--------	--------------

No gate pass requests found.

Welcome, Mahak Varshney!



Your Room
Room A-101



Your Roll No
104

Available Products

Notebook
Lined notebook for class notes
₹60.00
Add

Pen Set
Set of blue and black pens
₹30.00
Add

Eraser
Rubber eraser
₹50.00
Add

Pencil
HB pencil
₹25.00
Add

Marker
Permanent marker
₹100.00
Add

Ruler
30cm plastic ruler
₹50.00
Add

Sharpener

Pencil sharpener

₹75.00

 Add

Glue Stick

20g glue stick

₹20.00

 Add

Highlighter

Yellow highlighter pen

₹50.00

 Add

Sketch Book

A4 sheets (50)

₹250.00

 Add

Your Gate Passes

[Request New Gate Pass](#)

You have no gate pass requests.

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 

Submit Complaint

Title

Describe issue

Submit

My Complaints

Title	Description	Status	Submitted On
No complaints submitted yet.			

```

1 from extensions import db
2 from werkzeug.security import generate_password_hash, check_password_hash
3 from datetime import datetime
4
5 # All classes will inherit from db.Model
6 class Admin(db.Model):
7     id = db.Column(db.Integer, primary_key=True)
8     username = db.Column(db.String(80), unique=True, nullable=False)
9     password_hash = db.Column(db.String(128))
10
11     def set_password(self, password):
12         self.password_hash = generate_password_hash(password)
13
14     def check_password(self, password):
15         return check_password_hash(self.password_hash, password)
16
17 class Staff(db.Model):
18     id = db.Column(db.Integer, primary_key=True)
19     name = db.Column(db.String(100), nullable=False)
20     email = db.Column(db.String(120), unique=True, nullable=False)
21     department = db.Column(db.String(100))
22     password_hash = db.Column(db.String(128))
23
24     def set_password(self, password):
25         self.password_hash = generate_password_hash(password)
26
27     def check_password(self, password):
28         return check_password_hash(self.password_hash, password)
29
30
31 class Room(db.Model):
32     id = db.Column(db.Integer, primary_key=True)
33     room_no = db.Column(db.String(10), unique=True, nullable=False)
34     capacity = db.Column(db.Integer, nullable=False)
35     occupied = db.Column(db.Integer, default=0)
36     students = db.relationship('Student', back_populates='room') # link to students
37
38
39 class Student(db.Model):
40     id = db.Column(db.Integer, primary_key=True)
41     roll = db.Column(db.String(20), unique=True, nullable=False)
42     name = db.Column(db.String(100), nullable=False)
43     email = db.Column(db.String(100), nullable=False)
44     password_hash = db.Column(db.String(128))
45     room_id = db.Column(db.Integer, db.ForeignKey('room.id')) # FK to Room
46     room = db.relationship('Room', back_populates='students') # back-reference
47     last_active = db.Column(db.DateTime, nullable=True)
48     last_active_ip = db.Column(db.String(45), nullable=True)
49
50     is_assigned = db.Column(db.Boolean, default=False)
51
52     def set_password(self, password):
53         self.password_hash = generate_password_hash(password)
54
55     def check_password(self, password):
56         return check_password_hash(self.password_hash, password)
57

```

```

57
58 class Product(db.Model):
59     id = db.Column(db.Integer, primary_key=True)
60     name = db.Column(db.String(100), nullable=False)
61     description = db.Column(db.String(255))
62     price = db.Column(db.Float, nullable=False)
63     stock = db.Column(db.Integer, nullable=False)
64     min_quantity = db.Column(db.Integer, default=1)
65
66
67 class Complaint(db.Model):
68     id = db.Column(db.Integer, primary_key=True)
69     student_id = db.Column(db.Integer, db.ForeignKey('student.id'))
70     title = db.Column(db.String(100))
71     description = db.Column(db.Text)
72     status = db.Column(db.String(50), default='Pending')
73     date_submitted = db.Column(db.DateTime, default=datetime.utcnow)
74
75
76     student = db.relationship("Student", backref="complaints")
77
78
79 class Cart(db.Model):
80     id = db.Column(db.Integer, primary_key=True)
81     student_id = db.Column(db.Integer, db.ForeignKey('student.id'), nullable=False)
82     product_id = db.Column(db.Integer, db.ForeignKey('product.id'), nullable=False)
83     quantity = db.Column(db.Integer, nullable=False)
84     student = db.relationship('Student', backref=db.backref('cart_items', lazy=True))
85     product = db.relationship('Product', backref=db.backref('cart_items', lazy=True))
86
87
88 class GatePass(db.Model):
89     id = db.Column(db.Integer, primary_key=True)
90     student_id = db.Column(db.Integer, db.ForeignKey('student.id'))
91     reason = db.Column(db.String(200), nullable=False)
92     leave_date = db.Column(db.Date, nullable=False)
93     return_date = db.Column(db.Date, nullable=False)
94     status = db.Column(db.String(20), default='Pending')
95     date_requested = db.Column(db.DateTime, default=datetime.utcnow)
96
97     student = db.relationship('Student', backref='gate_passes')
98
99
100 class Fees(db.Model):
101     id = db.Column(db.Integer, primary_key=True)
102     student_id = db.Column(db.Integer, db.ForeignKey('student.id'), nullable=False)
103     month = db.Column(db.String(20), nullable=False)
104     year = db.Column(db.Integer, nullable=False)
105     amount_due = db.Column(db.Float, default=0)
106     amount_paid = db.Column(db.Float, default=0)
107     status = db.Column(db.String(20), default='Pending')
108
109     student = db.relationship('Student', backref='fees')
110
111     @property
112     def balance(self):
113         return self.amount_due - self.amount_paid

```

```
113
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
123     student = db.relationship('Student', backref='orders')
124     product = db.relationship('Product', backref='orders')
```