```
import os

# Global Variables
students = {}
courses_available = ("Python", "Data Science", "Web Development", "AI & ML")
student_file = "students.txt"

# 1. Add Student
def add_student():
    try:
        student_id = input("Enter Student ID: ")
        if student_id in students:
            print("Student ID already exists.")
            return

        name = input("Enter Student Name: ")
        age = int(input("Enter Age: "))
        print("Available Courses:", courses_available)
        enrolled = input("Enter courses separated by commas: ").split(",")
        courses_enrolled = set([course.strip() for course in enrolled if course.strip() in courses_available])

        total_fees = float(input("Enter Total Fees: ₹"))
        fees_paid = float(input("Enter Fees Paid: ₹"))
        balance = total_fees - fees_paid

        student_data = {
            "Name": name,
            "Age": age,
            "Courses": courses_enrolled,
            "Total Fees": total_fees,
            "Fees Paid": fees_paid,
            "Balance": balance
        }

        students[student_id] = student_data
        save_to_file(student_id, student_data)
        print(f"Student {name} added successfully!")

    except ValueError:
        print("Invalid input! Please enter correct data types.")

# 2. File Handling - Save
def save_to_file(student_id, data):
    try:
        with open(student_file, "a") as f:
            f.write(f"{student_id}|{data['Name']}|{data['Age']}|{','.join(data['Courses'])}|{data['Total Fees']}|{
    except Exception as e:
        print("Error writing to file:", e)

# File Handling - Load
def load_from_file():
    if not os.path.exists(student_file):
        return
    try:
        with open(student_file, "r") as f:
            for line in f:
                student_id, name, age, courses, total, paid, balance = line.strip().split("|")
                students[student_id] = {
                    "Name": name,
                    "Age": int(age),
                    "Courses": set(courses.split(",")),
                    "Total Fees": float(total),
                    "Fees Paid": float(paid),
                    "Balance": float(balance)
                }
    except FileNotFoundError:
        print("students.txt not found.")

# 3. View All Students
def view_students():
    if not students:
        print("No records
        return

    for sid, data in students.items():
```

```
    for sid, data in students.items():
        print(f"\nID: {sid}")
        print(f"Name: {data['Name']}")
        print(f"Age: {data['Age']}")
        print(f"Courses: {', '.join(data['Courses'])}")
        print(f"Total Fees: ₹{data['Total Fees']:,}")
        print(f"Fees Paid: ₹{data['Fees Paid']:,}")
        print(f"Balance: ₹{data['Balance']:,}")

# 4. Update Student
def update_student():
    sid = input("Enter Student ID to update: ")
    if sid not in students:
        print("Student not found.")
        return
    try:
        name = input("Enter New Name: ")
        age = int(input("Enter New Age: "))
        print("Available Courses:", courses_available)
        enrolled = input("Enter courses separated by commas: ").split(",")
        courses_enrolled = set([course.strip() for course in enrolled if course.strip() in courses_available])
        total_fees = float(input("Enter Total Fees: ₹"))
        fees_paid = float(input("Enter Fees Paid: ₹"))
        balance = total_fees - fees_paid

        students[sid] = {
            "Name": name,
            "Age": age,
            "Courses": courses_enrolled,
            "Total Fees": total_fees,
            "Fees Paid": fees_paid,
            "Balance": balance
        }

        overwrite_file()
        print("Student record updated.")
    except ValueError:
        print("Invalid input type!")

# 5. Remove Student
def remove_student():
    sid = input("Enter Student ID to remove: ")
    if sid in students:
        del students[sid]
        overwrite_file()
        print("Student record removed.")
    else:
        print("Student not found.")

# Overwrite the full file with current dictionary
def overwrite_file():
    try:
        with open(student_file, "w") as f:
            for sid, data in students.items():
                f.write(f"{sid}|{data['Name']}|{data['Age']}|{','.join(data['Courses'])}|{data['Total Fees']}|{dat
    except Exception as e:
        print("Error updating file:", e)

# 6. Generate Fee Report
def generate_fee_report():
    print("\nStudents with Pending Fees:")
    found = False
    for sid, data in students.items():
        if data['Balance'] > 0:
            found = True
            print(f"ID: {sid} | Name: {data['Name']} | Balance: ₹{data['Balance']:,}")
    if not found:
        print("All students have cleared their fees.")

# Main Menu
def menu():
    load_from_file()
    while True:
        print("\n========= Student Course Management System =========")
        print("1. Enroll a Student")
        print("2. View All Student Records")
        print("3. Update Student Information")
        print("4. Remove a Student Record")
        print("5. Generate Fee Report")
        print("6. Exit")

        choice = input("Enter your choice (1-6): ")
        if choice == "1":
            add_student()
        elif choice == "2":
            view_students()
        elif choice == "3":
            update_student()
        elif choice == "4":
            remove_student()
        elif choice == "5":
            generate_fee_report()
        elif choice == "6":
            print("Exiting program.")
            break
        else:
            print("Invalid choice. Try again.")

# Run the application
menu()
```

```
========= Student Course Management System =========
1. Enroll a Student
2. View All Student Records
3. Update Student Information
4. Remove a Student Record
5. Generate Fee Report
6. Exit
Enter your choice (1-6): 5

Students with Pending Fees:
All students have cleared their fees.

========= Student Course Management System =========
1. Enroll a Student
2. View All Student Records
3. Update Student Information
4. Remove a Student Record
5. Generate Fee Report
6. Exit
Enter your choice (1-6): 5

Students with Pending Fees:
All students have cleared their fees.

========= Student Course Management System =========
1. Enroll a Student
2. View All Student Records
3. Update Student Information
4. Remove a Student Record
5. Generate Fee Report
6. Exit
Enter your choice (1-6): 1
Enter Student ID: 1001
Enter Student Name: kishore
Enter Age: 21
Available Courses: ('Python', 'Data Science', 'Web Development', 'AI & ML')
Enter courses separated by commas: python,java,cloud,c++
Enter Total Fees: ₹2000000
Enter Fees Paid: ₹200000
Student kishore added successfully!

========= Student Course Management System =========
1. Enroll a Student
2. View All Student Records
3. Update Student Information
4. Remove a Student Record
5. Generate Fee Report
6. Exit
Enter your choice (1-6):
```

Start coding or generate with AI.