

# Chichebendu Blessed Umeh - What I Built for This Project

**\*\*My Role\*\*:** Email & Integration Specialist

**\*\*Project\*\*:** School Activity Booking System

---

## Quick Summary - What I Did

**\*\*In Simple Words\*\*:** I made the system talk to people and create documents.

**\*\*Think of it like this\*\*:**

- I'm the postman who delivers letters (emails)
- I'm the secretary who types up official invoices (PDFs)
- I'm the personal assistant who puts meetings in your calendar

**\*\*My Main Jobs\*\*:**

- Sending confirmation emails to parents
- Attaching calendar invites so you don't forget
- Creating professional PDF invoices
- Building the portal for Tutors to check attendance

---

## Part 1: The Email System

### ***What I Did (Simple Summary)***

- Made the website send real emails
- When you book, you get an email instantly
- It's not just text - it looks professional with colors and logos

### ***How Does It Work? (Easy Explanation)***

**\*\*Imagine writing a letter\*\*:**

1. You write the letter
2. You put it in an envelope
3. You put a stamp on it
4. You put it in the mailbox
5. The post office delivers it

**\*\*My code does exactly this, but super fast\*\*:**

**\*\*Step 1: Writing the Letter (The Template)\*\***

- I created a "fill-in-the-blanks" letter
- "Dear [Parent Name], you booked [Activity Name]..."
- Computer fills in the blanks automatically

**\*\*Step 2: The Post Office (SMTP Server)\*\***

- We use Gmail as our post office
- My code knocks on Gmail's door
- Says: "Here is a letter, please deliver it to parent@gmail.com"
- Gmail says: "Sure!" and sends it

### ***The Code (With Simple Explanation)***

```
def send_email(booking):  
    # Step 1: Write the letter  
    msg = Message(  
        subject="Booking Confirmed!",  
        recipients=[parent.email] # To: parent@gmail.com  
    )
```

```
# Step 2: Make it look pretty (HTML)
msg.html = """
Greenwood School
You have booked Swimming!
"""

# Step 3: Send it!
mail.send(msg)
**Real Example**:
Parent clicks "Book"
Computer instantly:
1. Grabs parent's email
2. Fills in the template
3. Connects to Gmail
4. Sends email
5. Parent's phone goes "Ding! You have mail!" ■
---
```

## Part 2: Calendar Invites (.ics files)

### ***What I Did (Simple Summary)***

- Made a file that adds the event to your phone's calendar
- You click it, and it saves the date/time automatically
- It even sets a reminder 24 hours before!

### ***How Does It Work? (Easy Explanation)***

**\*\*Imagine a digital business card\*\***:

- You tap phones, and the contact saves
- My file is like that, but for events

**\*\*The "ICS" File\*\***:

- It's a special text file that Calendar apps understand
- It speaks a special language called "iCalendar"

**\*\*What the file says (translated to human)\*\***:

```
BEGIN EVENT
DATE: December 15th
TIME: 3:00 PM
WHAT: Swimming Class
WHERE: School Pool
REMINDER: 1 day before
END EVENT
```

### ***The Code (With Simple Explanation)***

```
def generate_calendar_file(booking):
# Create the special text
content = f"""
BEGIN:VCALENDAR
BEGIN:VEVENT
DTSTART:{booking_date}T150000
SUMMARY:{activity_name}
END:VEVENT
END:VCALENDAR
"""

return content
**Real Example**:
1. You get the email
2. There is a file attached: "booking.ics"
3. You click it
```

4. Your iPhone/Android asks: "Add to Calendar?"
  5. You say "Yes"
  6. ■ Saved! Now your phone will remind you!
- 

## Part 3: PDF Invoice Generator

### *What I Did (Simple Summary)*

- Made the computer draw a professional invoice
- It's not just a webpage - it's a real PDF file you can download/print
- Looks like an official bank document

### *How Does It Work? (Easy Explanation)*

**\*\*Imagine a robot artist\*\*:**

- I tell the robot: "Draw a line here", "Write text here", "Put logo here"
- The robot draws it perfectly on a blank PDF page

**\*\*My Instructions to the Robot\*\*:**

1. "Start at top left"
2. "Draw School Logo"
3. "Move down 2 inches"
4. "Draw a table with 4 columns"
5. "Write 'Swimming Class' in column 1"
6. "Write '£25.00' in column 4"
7. "Draw a blue line at the bottom"

### *The Code (With Simple Explanation)*

```
def create_invoice(booking):
# Create blank PDF
pdf = SimpleDocTemplate("invoice.pdf")
# Create list of things to draw
elements = []
# Add Title
elements.append(Paragraph("INVOICE", big_font))
# Add Table
data = [['Activity', 'Price'], ['Swimming', '£25']]
table = Table(data)
elements.append(table)
# Build the PDF
pdf.build(elements)
**Real Example**:
Parent clicks "Download Invoice"
Computer:
1. Gets blank paper (digital)
2. Draws logo, address, details
3. Calculates total (£25)
4. Saves as "invoice_123.pdf"
5. Sends file to parent
Parent prints it for their records! ■
```

---

## Part 4: Tutor Portal & Attendance

### *What I Did (Simple Summary)*

- Built a special website section just for teachers
- They can see who is in their class
- They can mark attendance (Present/Absent)

## ***How Does It Work? (Easy Explanation)***

**\*\*Imagine a digital class register\*\*:**

- In old days: Teacher has paper list, ticks boxes with pen
- My system: Teacher has iPad, taps buttons on screen

**\*\*The Tutor's View\*\*:**

1. Login (using Sanchit's login system)
2. See list of classes ("Swimming", "Math", "Art")
3. Click "Swimming"
4. See list of kids ("John", "Emma", "Mike")
5. Tap "Present" next to John
6. Click "Save"

## ***The Code (With Simple Explanation)***

```
@app.route('/mark_attendance', methods=['POST'])
def mark_attendance():
    # Get the list from teacher
    student_id = request.form.get('student_id')
    status = request.form.get('status') # "Present" or "Absent"
    # Save to database
    attendance = Attendance(student=student_id, status=status)
    db.session.add(attendance)
    db.session.commit()
**Real Example**:
Teacher (Mr. Smith):
- Opens website on tablet
- Sees "Swimming Class - 3:00 PM"
- Sees list of 15 students
- Marks 14 as "Present"
- Marks 1 as "Absent" (Sick)
- Clicks Submit
Database:
- Updates instantly!
- Admin can see immediately that 1 student is absent.
---
```

## **My Contribution Summary**

**\*\*Files I Created/Modified\*\*:**

1. `app.py` - Added email, PDF, and tutor functions (250+ lines)
2. `templates/tutor/` - The teacher's website pages
3. Email Templates - The HTML designs for emails

**\*\*What Each Part Does (Simple)\*\*:**

Part	What It Does	Like...
Email System	Sends confirmations	Digital Postman
Calendar (.ics)	Adds event to phone	Personal Assistant
PDF Generator	Creates official bill	Robot Accountant
Tutor Portal	Teacher's dashboard	Digital Class Register
Attendance	Tracks who is present	Digital Roll Call

---

## Why This Matters

**\*\*Without my work\*\*:**

- ■ Parents wouldn't know if booking worked
- ■ Parents would forget the date (no calendar)
- ■ No proof of payment (no invoice)
- ■ Teachers wouldn't know who is coming

**\*\*With my work\*\*:**

- ■ Instant confirmation "Ding!" ■
- ■ Automatic reminders ■
- ■ Professional receipts ■
- ■ Organized classes ■■■

---

**\*\*Chichebendu Blessed Umeh\*\***

Email & Integration Specialist  
University of East London  
December 2025