**Form Builder Management System**

**Abstract:**

Data collection is an essential process in various domains, from surveys to user registrations. This project aims to develop a web-based form submission system using **Flask and MySQL**, where users can submit input, which is stored securely in a database. An admin panel enables authorized users to log in and manage the collected responses. The system provides **a structured, efficient, and secure alternative to existing online form solutions like Google Forms**, ensuring data privacy and accessibility through role-based authentication.

**Existing System:**

Many organizations rely on platforms like **Google Forms**, which store data on third-party cloud services, leading to concerns over **data privacy and limited customization**. These platforms often lack **role-based access control** and **advanced data management capabilities**.

**Proposed System:**

The proposed system is a **Flask-based web application with MySQL** as the database backend. It allows users to submit data through customizable web forms while providing **an admin dashboard for viewing, filtering, and managing responses**. The system ensures **secure authentication, data encryption, and structured storage**, making it an ideal alternative for organizations requiring **self-hosted** form solutions.

**Objective:**

* Develop a **web-based form submission** system.
* Store user inputs securely in a **MySQL database**.
* Implement **admin authentication** to access and manage collected data.
* Ensure **data privacy** by using encryption and role-based access.
* Provide a **customizable form interface** for different data collection needs.

**Methodology:**

1. **Frontend Development:**
   * HTML, CSS (**Tailwind CSS**) for form UI.
   * JavaScript for client-side validation.
2. **Backend Development:**
   * Flask for handling form submissions and admin authentication.
   * Flask-WTF for form handling with validation.
3. **Database Management:**
   * MySQL database to store form responses.
   * SQLAlchemy for ORM-based interaction.
4. **User Authentication:**
   * Admin login using Flask-Login.
   * Password hashing with Flask-Bcrypt for security.
5. **Admin Panel:**
   * Dashboard to **view, filter, and export** user-submitted data.
   * Pagination and sorting for efficient management.
6. **Security Measures:**
   * **Encryption** for sensitive data.
   * **Role-based access control** (Users submit data, Admin manages it).

**Conclusion:**

The **Flask-MySQL form submission system** provides a **secure, customizable, and efficient** alternative to cloud-based form solutions. With an intuitive user interface, a secure admin panel, and robust data management, it enhances **data privacy, security, and accessibility** while being **self-hosted and scalable**.

Source Code:

import os

from flask import Flask, render\_template, request, redirect, url\_for, flash

import pymysql

from werkzeug.utils import secure\_filename

from flask\_login import LoginManager, UserMixin, login\_user, login\_required, logout\_user, current\_user

app = Flask(\_\_name\_\_)

app.secret\_key = 'your\_secret\_key'

# Initialize Flask-Login

login\_manager = LoginManager()

login\_manager.init\_app(app)

login\_manager.login\_view = 'admin\_login'  # Redirect to login page if not authenticated

# Dummy User Database

users = {'admin': {'password': '1234'}}

# Configure upload folder

UPLOAD\_FOLDER = "static/uploads"

if not os.path.exists(UPLOAD\_FOLDER):

    os.makedirs(UPLOAD\_FOLDER)

app.config["UPLOAD\_FOLDER"] = UPLOAD\_FOLDER

# Connect to MySQL Database

db = pymysql.connect(

    host='localhost',

    user='root',

    password='1234',

    database='formsdb'

)

def fetch\_data(table\_name):

    cursor = db.cursor()

    cursor.execute(f"SELECT \* FROM {table\_name}")

    data = cursor.fetchall()

    cursor.close()

    return data

class User(UserMixin):

    def \_\_init\_\_(self, username):

        self.id = username

@login\_manager.user\_loader

def load\_user(user\_id):

    return User(user\_id) if user\_id in users else None

@app.route('/')

def index():

    return render\_template('index.html')

@app.route('/admin\_login', methods=['GET', 'POST'])

def admin\_login():

    if request.method == 'POST':

        username = request.form['username']

        password = request.form['password']

        if username in users and users[username]['password'] == password:

            user = User(username)

            login\_user(user)

            return redirect(url\_for('admin'))

    return render\_template('admin\_login.html')

@app.route('/admin')

@login\_required

def admin():

    return render\_template('admin.html')

@app.route('/student\_registration')

def student\_registration():

    headers = ["ID","Full Name", "Email", "Phone Number", "Date of Birth", "Address", "Course", "College Name"]

    data = fetch\_data('student\_registration')  # Ensure `fetch\_data` is implemented correctly

    return render\_template('student\_registration.html', form={"headers": headers, "data": data})

@app.route('/event\_registration')

def event\_registration():

    headers = ["ID","Full Name", "Email", "Phone Number", "Event Name", "Ticket Type", "Payment Status"]

    data = fetch\_data('event\_registration')

    return render\_template('event\_registration.html', form={"headers":headers,"data":data})

@app.route('/employee\_registration')

def employee\_registration():

    headers = ["ID", "Full Name", "Email", "Phone Number", "Employee ID", "Department", "Joining Date"]

    data = fetch\_data('employee\_registration')

    return render\_template('employee\_registration.html', form={"headers":headers,"data":data})

@app.route('/workshop\_registration')

def workshop\_registration():

    headers = ["ID", "Full Name", "Email", "Phone Number", "Workshop Name", "Preferred Date", "Payment Status"]

    data = fetch\_data('workshop\_registration')

    return render\_template('workshop\_registration.html', form={"headers":headers,"data":data})

@app.route('/general\_contact')

def general\_contact():

    headers = ["ID", "Full Name", "Email", "Phone Number", "Message"]

    data = fetch\_data('general\_contact')

    return render\_template('general\_contact.html', form={"headers":headers,"data":data})

@app.route('/support\_ticket')

def support\_ticket():

    headers = ["ID", "Full Name", "Email", "Phone Number", "Issue Category", "Description", "Priority"]

    data = fetch\_data('support\_ticket')

    return render\_template('support\_ticket.html', form={"headers":headers,"data":data})

@app.route('/customer\_inquiry')

def customer\_inquiry():

    headers = ["ID", "Full Name", "Email", "Phone Number", "Product/Service", "Message"]

    data = fetch\_data('customer\_inquiry')

    return render\_template('customer\_inquiry.html', form={"headers":headers,"data":data})

@app.route('/online\_order')

def online\_order():

    headers = ["ID", "Full Name", "Email", "Phone Number", "Shipping Address", "Product Name", "Quantity", "Payment Method"]

    data = fetch\_data('online\_order')

    return render\_template('online\_order.html', form={"headers":headers,"data":data})

@app.route('/product\_checkout')

def product\_checkout():

    headers = ["ID", "Full Name", "Email", "Billing Address", "Shipping Address", "Product Details", "Payment Details"]

    data = fetch\_data('product\_checkout')

    return render\_template('product\_checkout.html',form={"headers":headers,"data":data})

@app.route('/subscribers')

def subscribers():

    headers = ["Id", "Full Name", "Email", "Subscription Plan","Created At"]

    data = fetch\_data('subscribers')

    return render\_template('subscription\_payment.html', form={"headers":headers,"data":data})

@app.route('/customer\_feedback')

def customer\_feedback():

    headers = ["ID", "Full Name", "Email", "Phone Number", "Product/Service", "Rating", "Comments"]

    data = fetch\_data('customer\_feedback')

    return render\_template('customer\_feedback.html', form={"headers":headers,"data":data})

@app.route('/employee\_satisfaction')

def employee\_satisfaction():

    headers = ["ID", "Full Name", "Email", "Department", "Job Role", "Satisfaction Rating", "Feedback"]

    data = fetch\_data('employee\_satisfaction')

    return render\_template('employee\_satisfaction.html', form={"headers":headers,"data":data})

@app.route('/market\_research')

def market\_research():

    headers = ["ID", "Full Name", "Email", "Age Group", "Gender", "Preferences", "Feedback"]

    data = fetch\_data('market\_research')

    return render\_template('market\_research.html', form={"headers":headers,"data":data})

@app.route('/job\_application')

def job\_application():

    headers = ["ID", "Full Name", "Email", "Phone Number", "Resume Upload", "Job Position", "Experience", "Cover Letter"]

    data = fetch\_data('job\_application')

    return render\_template('job\_application.html', form={"headers":headers,"data":data})

@app.route('/loan\_application')

def loan\_application():

    headers = ["ID", "Full Name", "Email", "Phone Number", "Loan Amount", "Employment Status"]

    data = fetch\_data('loan\_application')

    return render\_template('loan\_application.html', form={"headers":headers,"data":data})

@app.route('/scholarship\_application')

def scholarship\_application():

    headers = ["ID", "Full Name", "Email", "Phone Number", "Course Name", "University Name", "Academic Achievements", "Personal Statement"]

    data = fetch\_data('scholarship\_application')

    return render\_template('scholarship\_application.html', form={"headers":headers,"data":data})

@app.route('/student\_registration\_form', methods=['GET', 'POST'])

def student\_registration\_form():

    if request.method == 'POST':

        fullname = request.form['fullname']

        email = request.form['email']

        phone = request.form['phone']

        dob = request.form['dob']

        address = request.form['address']

        course = request.form['course']

        college = request.form['college']

        try:

            cursor = db.cursor()

            sql = "INSERT INTO student\_registration (full\_name, email, phone\_number, date\_of\_birth, address, course, college\_name) VALUES (%s, %s, %s, %s, %s, %s, %s)"

            cursor.execute(sql, (fullname, email, phone, dob, address, course, college))

            db.commit()

            cursor.close()

            flash("Student registered successfully!", "success")

            return redirect(url\_for('student\_registration\_form'))

        except Exception as e:

            db.rollback()

            flash(f"Error: {str(e)}", "danger")

    return render\_template('studentform.html')

@app.route('/event\_registration\_form', methods=['GET', 'POST'])

def event\_registration\_form():

    if request.method == 'POST':

        fullname = request.form['fullname']

        email = request.form['email']

        phone = request.form['phone']

        event = request.form['event']

        ticket = request.form['ticket']

        payment = request.form['payment']

        try:

            cursor = db.cursor()

            sql = """INSERT INTO event\_registration

                     (full\_name, email, phone\_number, event\_name, ticket\_type, payment\_status)

                     VALUES (%s, %s, %s, %s, %s, %s)"""

            cursor.execute(sql, (fullname, email, phone, event, ticket, payment))

            db.commit()

            cursor.close()

            flash("Registration successful!", "success")

            return redirect(url\_for('event\_registration\_form'))

        except Exception as e:

            db.rollback()

            flash(f"Error: {str(e)}", "danger")

    return render\_template('eventform.html')

@app.route('/employee\_registration\_form', methods=['GET', 'POST'])

def employee\_registration\_form():

    if request.method == 'POST':

        fullname = request.form['fullname']

        email = request.form['email']

        phone = request.form['phone']

        employee\_id = request.form['employee\_id']

        department = request.form['department']

        joining\_date = request.form['joining\_date']

        try:

            cursor = db.cursor()

            query = "INSERT INTO employee\_registration (full\_name, email, phone\_number, employee\_id, department, joining\_date) VALUES (%s, %s, %s, %s, %s, %s)"

            cursor.execute(query, (fullname, email, phone, employee\_id, department, joining\_date))

            db.commit()

            flash('Employee Registered Successfully!', 'success')

            return redirect(url\_for('employee\_registration\_form'))

        except Exception as e:

            db.rollback()

            flash(f'Error: {str(e)}', 'danger')

    return render\_template('employeeform.html')

@app.route('/workshop\_registration\_form', methods=['GET', 'POST'])

def workshop\_registration\_form():

    if request.method == 'POST':

        fullname = request.form['fullname']

        email = request.form['email']

        phone = request.form['phone']

        workshop\_name = request.form['workshop\_name']

        preferred\_date = request.form['preferred\_date']

        payment\_status = request.form['payment\_status']

        try:

            cursor = db.cursor()

            query = """

                INSERT INTO workshop\_registration

                (full\_name, email, phone\_number, workshop\_name, preferred\_date, payment\_status)

                VALUES (%s, %s, %s, %s, %s, %s)

            """

            cursor.execute(query, (fullname, email, phone, workshop\_name, preferred\_date, payment\_status))

            db.commit()

            flash('Workshop Registered Successfully!', 'success')

            return redirect(url\_for('workshop\_registration\_form'))

        except Exception as e:

            db.rollback()

            flash(f'Error: {str(e)}', 'danger')

    return render\_template('workshopform.html')

@app.route('/contact\_form', methods=['GET', 'POST'])

def contact\_form():

    if request.method == 'POST':

        fullname = request.form['fullname']

        email = request.form['email']

        phone = request.form['phone']

        message = request.form['message']

        try:

            cursor = db.cursor()

            query = """

                INSERT INTO general\_contact

                (full\_name, email, phone\_number, message)

                VALUES (%s, %s, %s, %s)

            """

            cursor.execute(query, (fullname, email, phone, message))

            db.commit()

            flash('Contact Registered Successfully!', 'success')

            return redirect(url\_for('contact\_form'))

        except Exception as e:

            db.rollback()

            flash(f'Error: {str(e)}', 'danger')

    return render\_template('generalcontact.html')

@app.route('/support\_ticket\_form', methods=['GET', 'POST'])

def support\_ticket\_form():

    if request.method == 'POST':

        fullname = request.form['fullname']

        email = request.form['email']

        phone = request.form['phone']

        issue = request.form['category']

        description = request.form['description']

        priority = request.form['priority']

        try:

            cursor = db.cursor()

            query = """

                INSERT INTO support\_ticket

                (full\_name, email, phone\_number, issue\_category, description, priority)

                VALUES (%s, %s, %s, %s, %s, %s)

            """

            cursor.execute(query, (fullname, email, phone, issue, description, priority))

            db.commit()

            flash('Support Ticket Given Successfully!', 'success')

            return redirect(url\_for('support\_ticket\_form'))

        except Exception as e:

            db.rollback()

            flash(f'Error: {str(e)}', 'danger')

    return render\_template('supportform.html')

@app.route('/submit\_inquiry', methods=['GET','POST'])

def submit\_inquiry():

    if request.method == 'POST':

        fullname = request.form['fullname']

        email = request.form['email']

        phone = request.form['phone']

        product\_service = request.form['product\_service']

        message = request.form['message']

        try:

            cursor = db.cursor()

            sql = """INSERT INTO customer\_inquiry (full\_name, email, phone\_number, product\_service, message)

                     VALUES (%s, %s, %s, %s, %s)"""

            cursor.execute(sql, (fullname, email, phone, product\_service, message))

            db.commit()

            flash('Your inquiry has been submitted successfully!', 'success')

            return redirect(url\_for('submit\_inquiry'))

        except Exception as e:

            db.rollback()

            flash(f'Error: {str(e)}', 'danger')

        finally:

            cursor.close()

    return render\_template('customerEnquiryform.html')

@app.route('/submit\_order', methods=['GET','POST'])

def submit\_order():

    if request.method == 'POST':

        fullname = request.form['fullname']

        email = request.form['email']

        phone = request.form['phone']

        address = request.form['address']

        product = request.form['product']

        quantity = request.form['quantity']

        payment\_method = request.form['payment\_method']

        try:

            cursor = db.cursor()

            sql = """INSERT INTO online\_order (full\_name, email, phone\_number, shipping\_address, product\_name, quantity, payment\_method)

                     VALUES (%s, %s, %s, %s, %s, %s, %s)"""

            cursor.execute(sql, (fullname, email, phone, address, product, quantity, payment\_method))

            db.commit()

            flash('Your order has been placed successfully!', 'success')

            return redirect(url\_for('submit\_order'))

        except Exception as e:

            db.rollback()

            flash(f'Error: {str(e)}', 'danger')

        finally:

            cursor.close()

    return render\_template('onlineorderform.html')

@app.route('/checkout', methods=['GET', 'POST'])

def checkout():

    if request.method == 'POST':

        fullname = request.form['full\_name']

        email = request.form['email']

        billing\_address = request.form['billing\_address']

        shipping\_address = request.form['shipping\_address']

        product\_details = request.form['product\_details']

        payment\_details = request.form['payment\_details']

        try:

            cursor = db.cursor()

            sql = """INSERT INTO product\_checkout (full\_name, email, billing\_address , shipping\_address, product\_details, payment\_details)

                        VALUES (%s, %s, %s, %s, %s, %s)"""

            cursor.execute(sql, (fullname, email, billing\_address, shipping\_address, product\_details, payment\_details))

            db.commit()

            flash('Your product will be delivered successfully!', 'success')

            return redirect(url\_for('checkout'))

        except Exception as e:

            db.rollback()

            flash(f'Error: {str(e)}', 'danger')

        finally:

            cursor.close()

    return render\_template('productcheckoutform.html')

@app.route('/subscribe', methods=['GET', 'POST'])

def subscribe():

    if request.method == 'POST':

        full\_name = request.form['full\_name']

        email = request.form['email']

        subscription\_plan = request.form['subscription\_plan']

        try:

            cursor = db.cursor()

            sql = "INSERT INTO subscribers (full\_name, email, subscription\_plan) VALUES (%s, %s, %s)"

            cursor.execute(sql, (full\_name, email, subscription\_plan))

            db.commit()

            flash('Your product will be delivered successfully!', 'success')

            return redirect(url\_for('subscribe'))

        except Exception as e:

            db.rollback()

            flash(f'Error: {str(e)}', 'danger')

        finally:

            cursor.close()

    return render\_template('subscriptionpayment.html')

@app.route("/submit\_feedback", methods=['GET',"POST"])

def submit\_feedback():

    if request.method == "POST":

        full\_name = request.form["full\_name"]

        email = request.form["email"]

        phone\_number = request.form["phone\_number"]

        product\_service = request.form["product\_service"]

        rating = request.form["rating"]

        comments = request.form["comments"]

        sql = """INSERT INTO customer\_feedback (full\_name, email, phone\_number, product\_service, rating, comments)

                 VALUES (%s, %s, %s, %s, %s, %s)"""

        values = (full\_name, email, phone\_number, product\_service, rating, comments)

        try:

            cursor = db.cursor()

            cursor.execute(sql, values)

            db.commit()

            flash('Thankyou for your feedback!', 'success')

            return redirect(url\_for('submit\_feedback'))

        except Exception as e:

            db.rollback()

            flash(f'Error: {str(e)}', 'danger')

        finally:

            cursor.close()

    return render\_template('feedback.html')

@app.route("/submit\_survey", methods=['GET',"POST"])

def submit\_survey():

    if request.method == "POST":

        full\_name = request.form["full\_name"]

        email = request.form["email"]

        department = request.form["department"]

        job\_role = request.form["job\_role"]

        satisfaction\_rating = request.form["satisfaction\_rating"]

        feedback = request.form["feedback"]

        sql = """INSERT INTO employee\_satisfaction (full\_name, email, department, job\_role, satisfaction\_rating, feedback)

                 VALUES (%s, %s, %s, %s, %s, %s)"""

        values = (full\_name, email, department, job\_role, satisfaction\_rating, feedback)

        try:

            cursor = db.cursor()

            cursor.execute(sql, values)

            db.commit()

            flash('Thankyou for your feedback!', 'success')

            return redirect(url\_for("submit\_survey"))

        except Exception as e:

            db.rollback()

            print("Error:", e)

    return render\_template('employeesatisfaction.html')

@app.route("/submit\_research", methods=['GET',"POST"])

def submit\_research():

    if request.method == "POST":

        full\_name = request.form["full\_name"]

        email = request.form["email"]

        age\_group = request.form["age\_group"]

        gender = request.form["gender"]

        preferences = request.form["preferences"]

        feedback = request.form["feedback"]

        sql = """INSERT INTO market\_research (full\_name, email, age\_group, gender, preferences, feedback)

                 VALUES (%s, %s, %s, %s, %s, %s)"""

        values = (full\_name, email, age\_group, gender, preferences, feedback)

        try:

            cursor = db.cursor()

            cursor.execute(sql, values)

            db.commit()

            flash('Thankyou for your feedback!', 'success')

            return redirect(url\_for('submit\_research'))

        except Exception as e:

            db.rollback()

            flash(f'Error: {str(e)}', 'danger')

        finally:

            cursor.close()

    return render\_template('marketresearch.html')

@app.route("/job\_application\_form", methods=['GET',"POST"])

def job\_application\_form():

    if request.method == "POST":

        full\_name = request.form["full\_name"]

        email = request.form["email"]

        phone\_number = request.form["phone\_number"]

        job\_position = request.form["job\_position"]

        experience = request.form["experience"]

        cover\_letter = request.form["cover\_letter"]

        # Handle file upload

        resume\_file = request.files["resume\_upload"]

        if resume\_file:

            UPLOAD\_FOLDER = "static/uploads"

            if not os.path.exists(UPLOAD\_FOLDER):

                os.makedirs(UPLOAD\_FOLDER)

            filename = secure\_filename(resume\_file.filename)

            resume\_path = os.path.join(app.config["UPLOAD\_FOLDER"], filename)

            resume\_file.save(resume\_path)

            sql = """INSERT INTO job\_application

                     (full\_name, email, phone\_number, resume\_upload, job\_position, experience, cover\_letter)

                     VALUES (%s, %s, %s, %s, %s, %s, %s)"""

            values = (full\_name, email, phone\_number, filename, job\_position, experience, cover\_letter)

            try:

                cursor = db.cursor()

                cursor.execute(sql, values)

                db.commit()

                flash('Your Application has been submitted', 'success')

                return redirect(url\_for('job\_application\_form'))

            except Exception as e:

                db.rollback()

                flash(f'Error: {str(e)}', 'danger')

            finally:

                cursor.close()

    return render\_template('jobapplication.html')

@app.route("/loan\_application\_form", methods=["GET", "POST"])

def loan\_application\_form():

    if request.method == "POST":

        full\_name = request.form["full\_name"]

        email = request.form["email"]

        phone\_number = request.form["phone\_number"]

        loan\_amount = request.form["loan\_amount"]

        employment\_status = request.form["employment\_status"]

        sql = """INSERT INTO loan\_application

                 (full\_name, email, phone\_number, loan\_amount, employment\_status)

                 VALUES (%s, %s, %s, %s, %s)"""

        values = (full\_name, email, phone\_number, loan\_amount, employment\_status)

        try:

            cursor = db.cursor()

            cursor.execute(sql, values)

            db.commit()

            flash("Your Loan Application has been submitted", "success")

            return redirect(url\_for("loan\_application\_form"))

        except Exception as e:

            db.rollback()

            flash(f"Error: {str(e)}", "danger")

        finally:

            cursor.close()

    return render\_template("loanapplication.html")

@app.route("/scholarship\_application\_form", methods=["GET", "POST"])

def scholarship\_application\_form():

    if request.method == "POST":

        full\_name = request.form["full\_name"]

        email = request.form["email"]

        phone\_number = request.form["phone\_number"]

        course\_name = request.form["course\_name"]

        university\_name = request.form["university\_name"]

        academic\_achievements = request.form["academic\_achievements"]

        personal\_statement = request.form["personal\_statement"]

        sql = """INSERT INTO scholarship\_application

                 (full\_name, email, phone\_number, course\_name, university\_name, academic\_achievements, personal\_statement)

                 VALUES (%s, %s, %s, %s, %s, %s, %s)"""

        values = (full\_name, email, phone\_number, course\_name, university\_name, academic\_achievements, personal\_statement)

        try:

            cursor = db.cursor()

            cursor.execute(sql, values)

            db.commit()

            flash("Your Scholarship Application has been submitted", "success")

            return redirect(url\_for("scholarship\_application\_form"))

        except Exception as e:

            db.rollback()

            flash(f"Error: {str(e)}", "danger")

        finally:

            cursor.close()

    return render\_template("scolarshipapplication.html")

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)















