

- 1) Write a flask program for display name, email, phone
 - From flask import Flask

```
App = Flask(__name__)
```

```
@app.route('/')
```

```
@app.route('/index/')
```

```
Def hello():
```

```
    Return '<h1>Hello, World!</h1>'
```

```
@app.route('/about/')
```

```
Def about():
```

```
    Return '<h3>This is a Flask web application.</h3>'
```

- Email sending in Flask-Mail is handled by an instance of a Mail class.

```
From flask import Flask
```

```
From flask_mail import Mail
```

```
App = Flask(app_name) # pick the name
```

```
Mail = Mail(app)
```

- 2) Write a flask program should cover cookies, sessions.

```
from flask import Flask, session
```

```
app = Flask(__name__)
```

```
app.config['SECRET_KEY'] = 'super secret'
```

```
@app.route('/start_counter')
```

```
defstart_counter():
```

```
    session['count'] = 0
```

```
    return 'New counter started!'
```

```
@app.route('/count')
```

```

def count():
    count = session.get('count')
    if isinstance(count, int):
        session['count'] += 1
        return str(session['count'])
    return 'No counter set!'

@app.route('/get_count')
def get_count():
    return str(session.get('count', 'No counter set!'))

@app.route('/clear_count')
def clear_count():
    session.pop('count', None)
    return 'Counter cleared!'

if __name__ == '__main__':
    app.run(debug=True)

```

Cookie:

```

From flask import Flask, render_template, request, redirect, url_for, flash, make_response
#...
@app.route('/cookie/')
def cookie():
    Res = make_response("Setting a cookie")
    Res.set_cookie('foo', 'bar', max_age=60*60*24*365*2)
    Return res

```

3) Write flask program which should display resume details & also upload

```

def allowed_file(filename):
    return '.' in filename and \
        filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

@app.route('/', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        # check if the post request has the file part
        if 'file' not in request.files:
            flash('No file part')
            return redirect(request.url)
        file = request.files['file']
        # If the user does not select a file, the browser submits an

```

```
# empty file without a filename.  
if file.filename == "":  
    flash('No selected file')  
    return redirect(request.url)  
if file and allowed_file(file.filename):  
    filename = secure_filename(file.filename)  
file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))  
    return redirect(url_for('download_file', name=filename))  
return ""
```

Batch members:

k.veeramani
s.anadhakumar
t.manikandan
s.venkatesh