```
render = web.template.render('templates/')

urls = (
    '/', 'index',
)

profile = open('profile.yaml')
profile_data = yaml.safe_load(profile)
profile.close()

# Functioning Variables
name = profile_data['name']

tts('Welcome ' + name + ', systems are now ready to run. How can I help you?')

class index:
    def GET(self):
        return render.index()

    def POST(self):
        x = web.input(myfile={})
        filedir = os.getcwd() + '/uploads' # change this to the directory
        you want to store the file in.
        if 'myfile' in x: # to check if the file-object is created
            filepath=x.myfile.filename.replace('\\','/') # replaces the
            windows-style slashes with linux ones.
            filename=filepath.split('/')[-1] # splits the command and
            chooses the last part (the filename with extension)
            fout = open(filedir +'/'+ filename,'w') # creates the directory
            where the uploaded file should be stored
            fout.write(x.myfile.file.read()) # writes the uploaded file to
            the newly created file.
            fout.close() # closes the file, upload complete.
        os.system('python main.py ' + filename)

if __name__ == "__main__":
    app = web.application(urls, globals())
    app.run()
```

You have to import the os, yaml, and web modules. You call the web.template.render() function because you are using web.py's templating engine. This function takes the location of the templates as an argument. Next you specify the list of URLs used in the application. You then define the index class, which contains the back-end code for the index page. The GET() function handles the rendering of the index page.

The POST() function handles the file upload using web.py's form-handling technology and saves the upload to the uploads folder. The comments explain the functionality of each line of code in the function. Finally, the file that is uploaded by the