**Note the following for:  
  
Admin Data:  
When you run app.py first time, the app will auto-create a default admin:**

* **Email: admin@example.com**
* **Password: Admin@1234**
* **Username: system.admin.sou.min**

**Pseudo Code and Flow Charts of the main functioning of the app:**

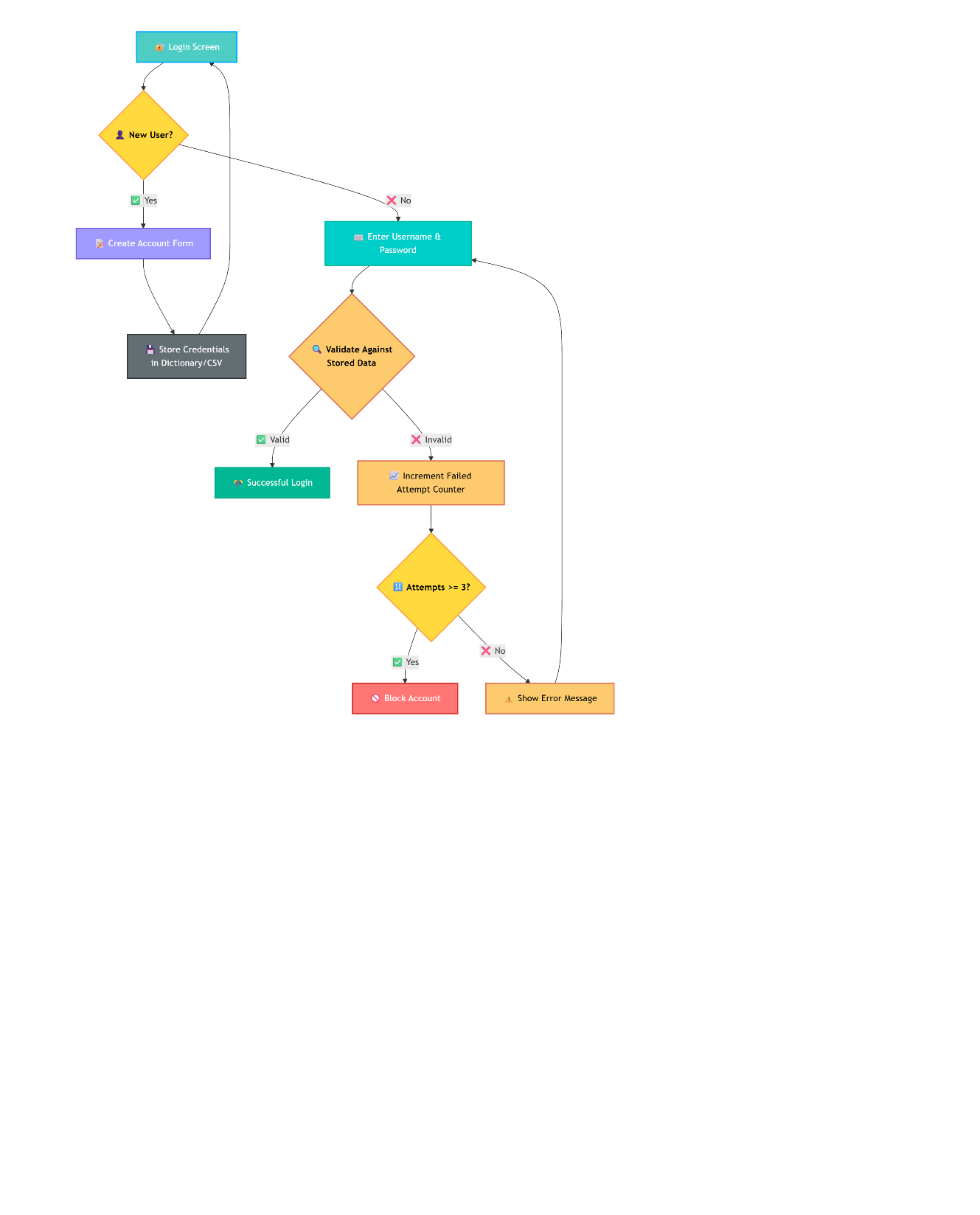
## **Alogrithm Sytem Overview Diagrams:**

### **App flow**

A diagram of a flowchart

AI-generated content may be incorrect.

### **User Login Flow**



### **Dashboard Redirection Flow**

A diagram of a diagram

AI-generated content may be incorrect.

### **Password Reset Flow**

A diagram of a flowchart

AI-generated content may be incorrect.

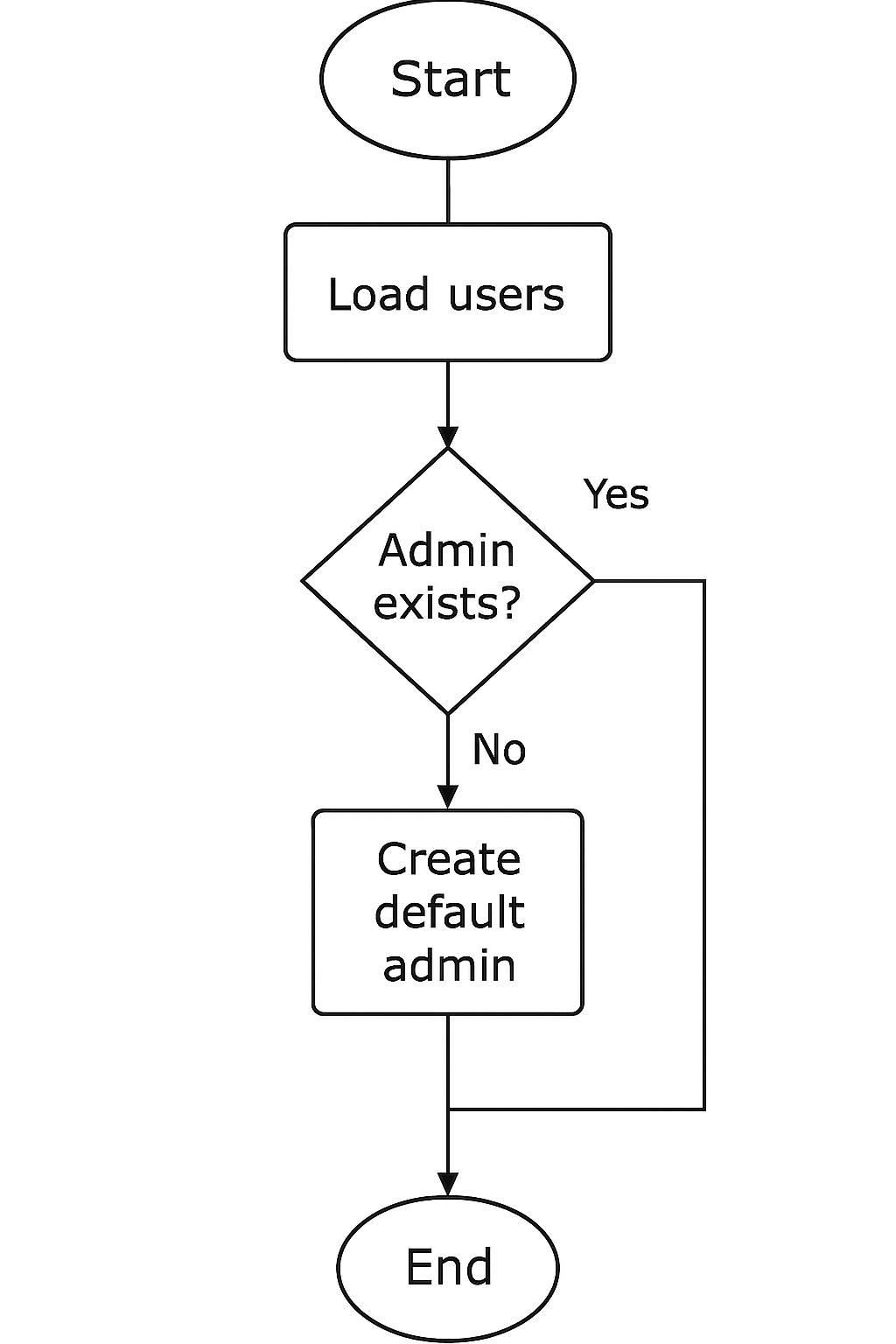
### **Administrator Unlock Flow**

A diagram of a software system

AI-generated content may be incorrect.

## **PSEUDOCODE AND FLOWCHARTS FOR ALL FUNCTIONS:**

### **create\_initial\_admin()**



FUNCTION create\_initial\_admin():

users\_list = load\_users().get("users", [])

IF no user with role == "Administrator" THEN

admin = create\_user(

first="System",

last="Admin",

email="admin@example.com",

country="SouthAfrica",

organization="MINN",

role="Administrator",

password="Admin@1234"

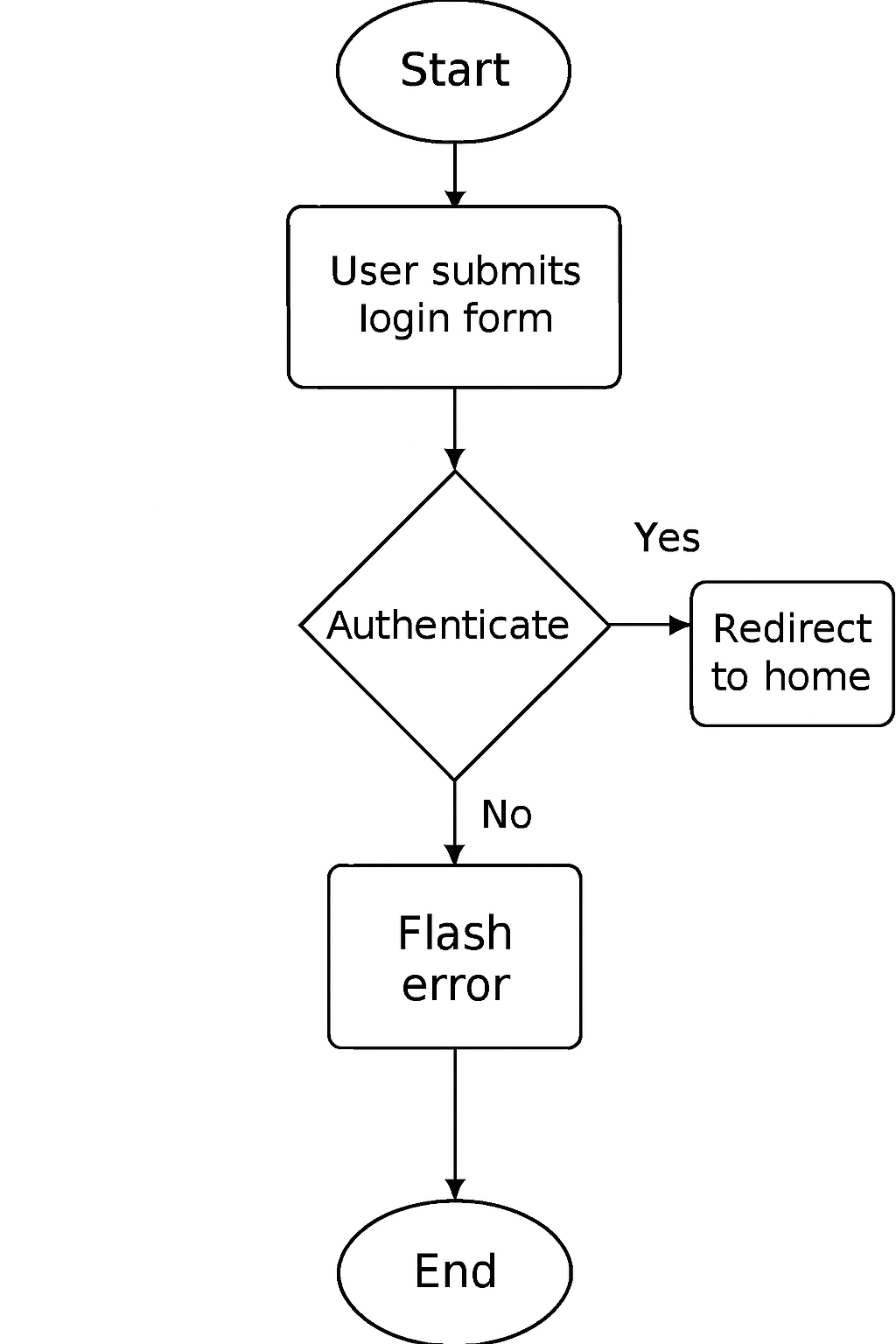
)

PRINT "Default admin created"

ENDIF

END FUNCTION

### **login()**



ROUTE /login (methods = GET, POST):

IF request.method == POST THEN

username = form.username

password = form.password

success, message = authenticate(username, password)

IF success THEN

flash("Login successful")

redirect to home

ELSE

flash(message)

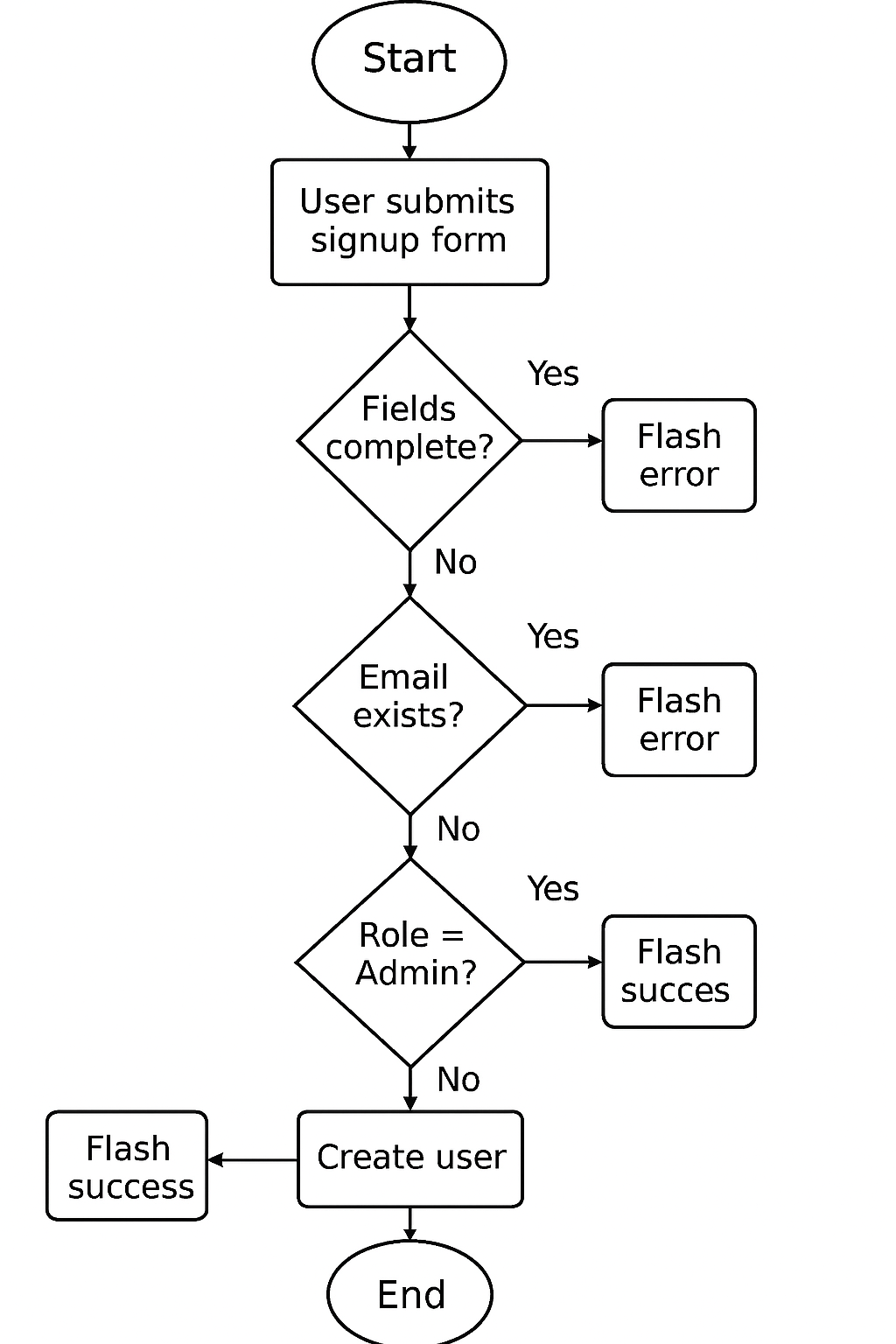
ENDIF

ELSE

render login.html

ENDIF

### **signup()**



ROUTE /signup (methods = GET, POST):

IF request.method == POST THEN

Collect form inputs

IF any field missing THEN

flash("Missing fields")

return signup.html

ENDIF

IF password != confirm THEN

flash("Passwords do not match")

return signup.html

ENDIF

IF find\_user\_by\_email(email) THEN

flash("Email already exists")

return signup.html

ENDIF

IF role == "Administrator" THEN

flash("Admins cannot self-register")

return signup.html

ENDIF

new\_user = create\_user(first, last, email, country, org, role, password)

flash("Account created. Please login.")

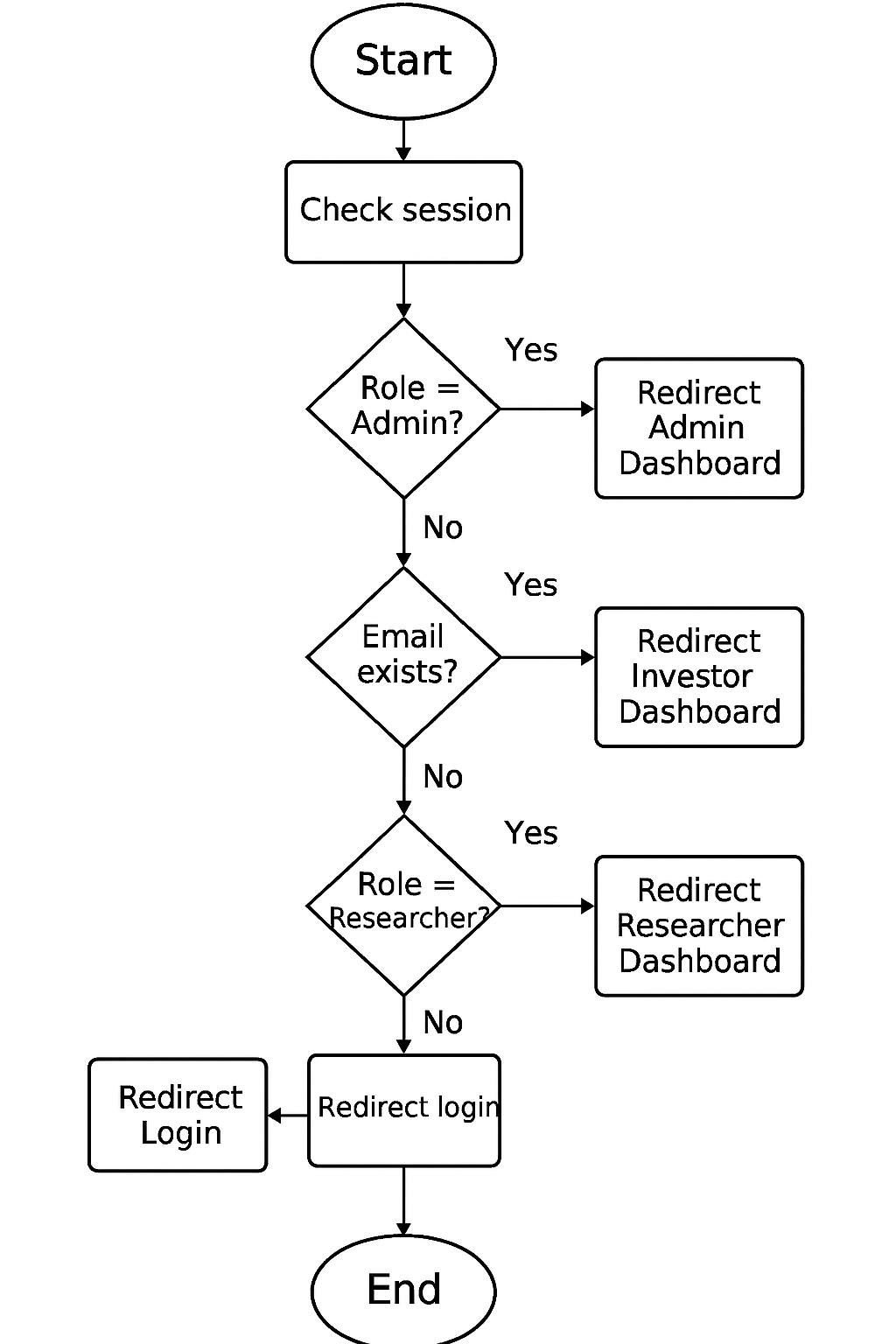
redirect to /login

ELSE

render signup.html

ENDIF

### **dashboard\_admin()**



ROUTE /dashboard/admin:

REQUIRE ROLE = "Administrator"

users = load\_users()

role\_counts = {"Administrator":0, "Investor":0, "Researcher":0}

FOR each user IN users:

role\_counts[user.role] += 1

ENDFOR

render dashboard\_admin.html(users, role\_counts)

### **dashboard\_investor()**

ROUTE /dashboard/investor:

REQUIRE ROLE = "Investor"

minerals = load\_minerals\_json()

chart\_html = generate\_overview\_chart()

map\_path = generate\_africa\_mineral\_map(minerals)

render dashboard\_investor.html(chart\_html, map\_path)

### **dashboard\_researcher()**

ROUTE /dashboard/researcher:

REQUIRE ROLE = "Researcher"

minerals = load\_minerals\_json()

map\_path = generate\_africa\_mineral\_map(minerals)

render dashboard\_researcher.html(minerals, map\_path)

### **reset\_request()**

ROUTE /reset-request (GET, POST):

IF POST:

email = form.email

user = find\_user\_by\_email(email)

IF user EXISTS:

token = generate\_reset\_token(email)

reset\_link = url\_for('reset\_password\_route', token)

flash("Link generated: " + reset\_link)

ELSE:

flash("If account exists, link sent.")

redirect to login

ELSE:

render reset\_request.html

### **reset\_password\_route(token)**

ROUTE /reset/<token> (GET, POST):

email = verify\_reset\_token(token)

IF NOT email:

flash("Invalid token")

redirect to /reset-request

IF POST:

pw = form.password

confirm = form.confirm

IF pw != confirm:

flash("Passwords do not match")

return reset\_password.html

reset\_password(email, pw)

flash("Password reset. Please login.")

redirect to login

ELSE:

render reset\_password.html

### **admin\_unlock(user\_id)**

ROUTE /admin/unlock/<user\_id> (POST):

REQUIRE ROLE = "Administrator"

ok = unlock\_account(user\_id)

IF ok:

flash("Account unlocked.")

ELSE:

flash("User not found.")

redirect to /dashboard/admin

### **minerals\_dashboard()**

ROUTE /minerals:

minerals = load\_minerals\_json()

chart\_html = generate\_overview\_chart()

render minerals\_dashboard.html(minerals, chart\_html)

### **mineral\_detail\_chart(mineral\_name)**

ROUTE /minerals/<mineral\_name>:

chart\_html = generate\_mineral\_chart(mineral\_name)

render minerals\_dashboard.html(chart\_html, single=True)