NSC Assignment 4

Project 1

Arpan Kumar (2021020)
Pranav Tanwar(2022368)

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Structure

- app.py: Replicates the university server, manages registrations and credentials, generates the RSA keys for signing, and issues PDFs
- user_app.py: Replicates the student server, allows logins and registrations, requests documents from university server and verifies them.
- rsa_utils.py: Implements RSA algorithm functions for encryption, decryption, and digital signatures, as well as SHA256 for verification.
- database.json: Stores the university details and the student details.

Secure retrieval of GMT Date/Time

- Method used in code: datetime.now().strftime(...) (from app.py, user_app.py)

```
timestamp = datetime.now().strftime("%Y%m%d%H%M%S")
filename = f"{doc_type}_{roll}_{timestamp}.pdf"
path = os.path.join(PDF_DIR, filename)
```

Source Reliability:

- Python's built-in datetime module is secure and reliable.
- No external network source used; hence secure by default.
- Security of Communication:
- No external requests made for obtaining time; local server time ensures inherent security.

Ensuring Secure Document Access

- Student Authentication (user_app.py):
- Students authenticated using a secure roll number and password combination.

```
def login(self, roll, pwd):

    s = self.db['university']['students'].get(roll)
    if s and s['password']==pwd:
        self.current_student = roll
        return True
    return False
```

Document Origin and Tracing

Signature in PDF (app.py):

- Signature embedded in each PDF, ensuring traceability:

```
c.setFont("Helvetica-Oblique", 8)
c.drawString(72, y, f"Signature: {signature}")
```

Public-Key Access and Management

- Public Key Necessity: Yes, for verifying digital signatures.
- University's Public Key Endpoint (app.py):

```python
@app.route('/api/public_key', methods=['GET'])
def get_public_key():
    return jsonify({
        'public_key': university.db['university']['public_key']
    })
```

- Student Public Key Management (app.py, user_app.py):
- Public keys stored securely in database.json:

```
"students": {
  "101": {
    "name": "AK",
    "password": "1234",
    "public_key": [
      18547,
      49187
    ],
```

Security Properties Addressed

Confidentiality:

- Achieved via RSA encryption (rsa_utils.py):

```
def encrypt(pk, plaintext):
    """Encrypt the plaintext using public key."""
    key, n = pk
    # Convert text to number representation
    cipher = [pow(ord(char), key, n) for char in plaintext]
    return cipher
```

Authentication:

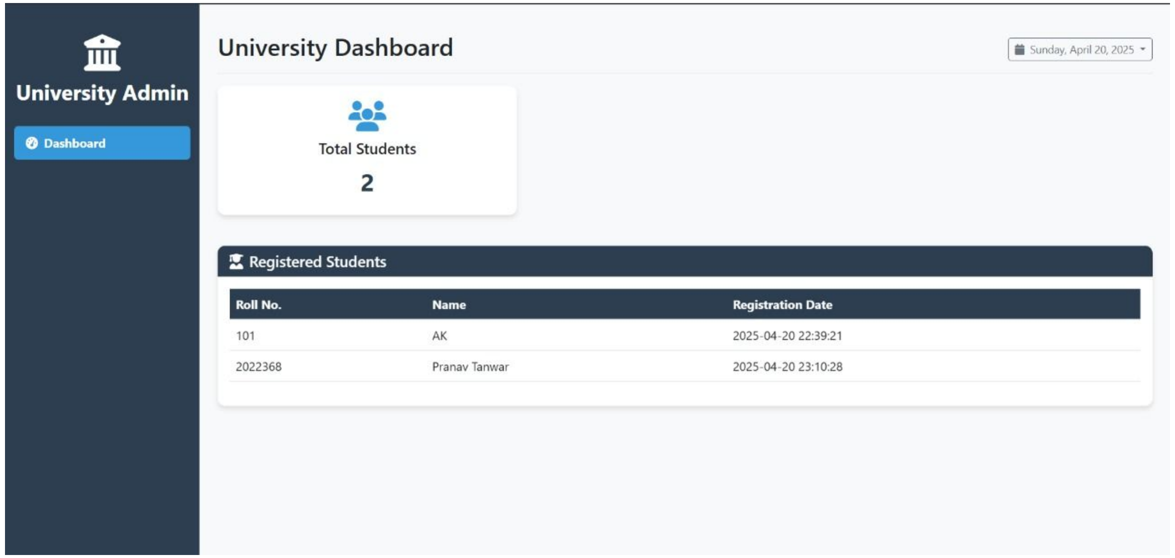
- Students and University authenticate securely using stored credentials and RSA key verification.

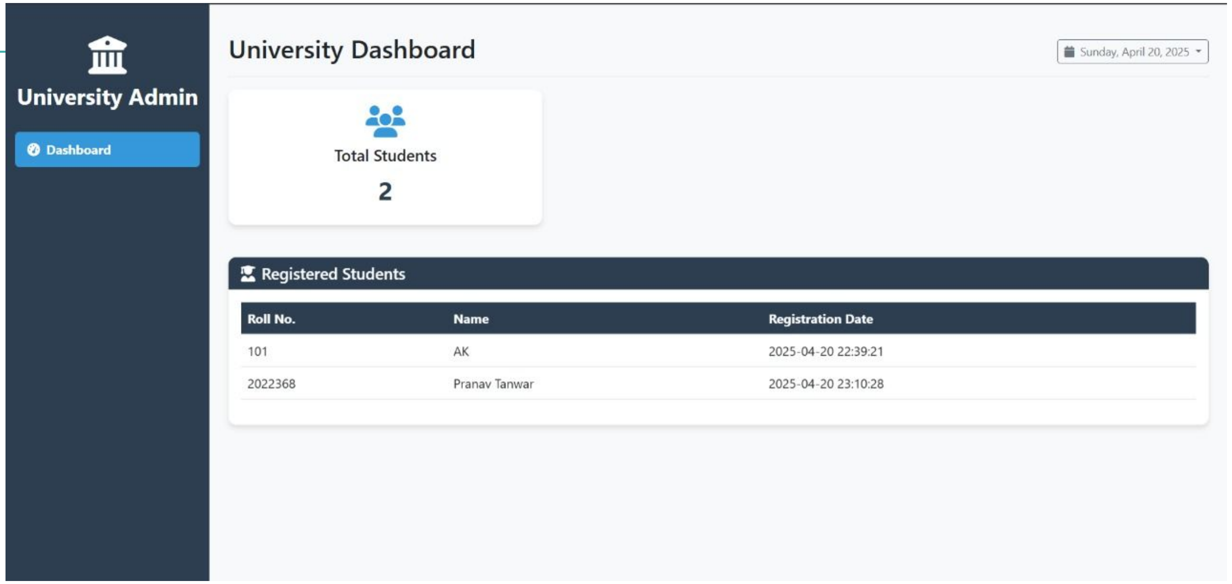
Integrity:

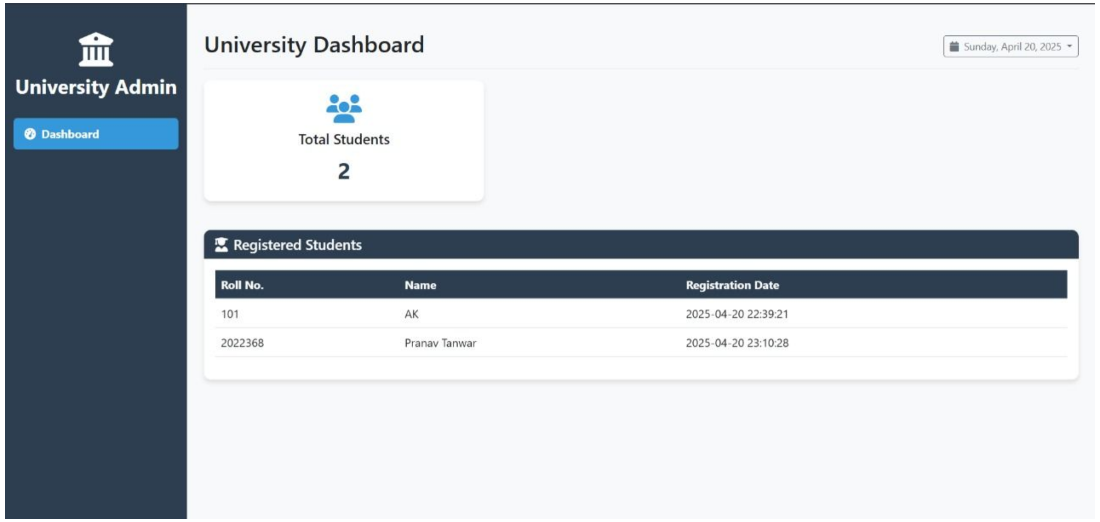
- SHA-256 hashing and digital signatures ensure document integrity:

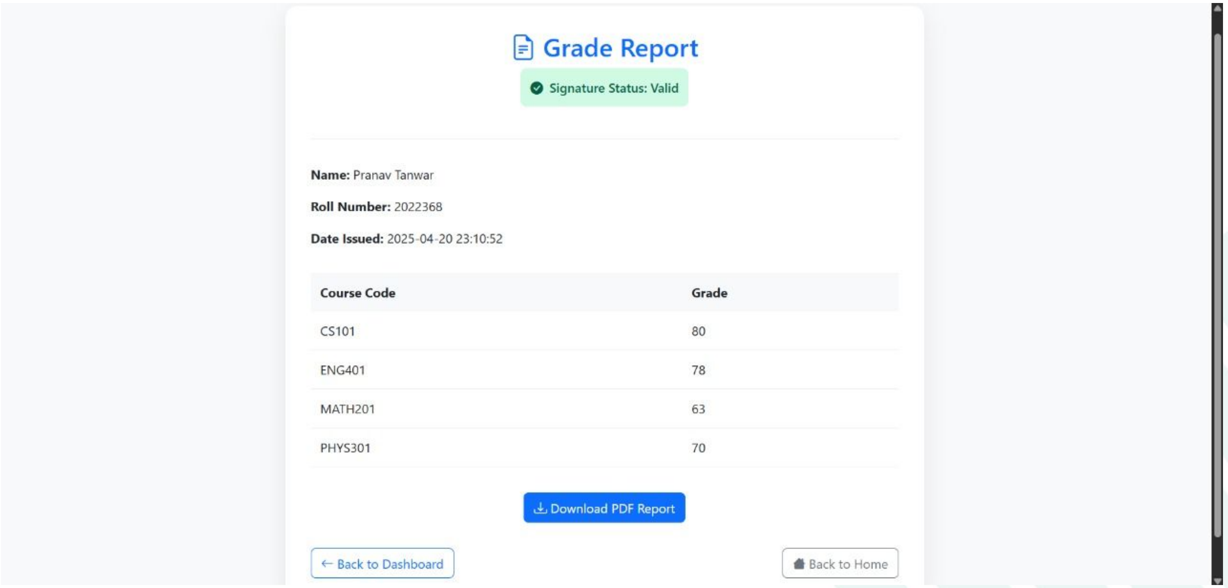
```
def sha256_hash(data):
    """Generate SHA-256 hash of the data."""
    if isinstance(data, str):
        data = data.encode('utf-8')
    return hashlib.sha256(data).hexdigest()
```

Frontend


University Admin
Dashboard
University Dashboard
Sunday, April 20, 2025
Total Students
2
Registered Students
Roll No.
Name
Registration Date
101
AK
2025-04-20 22:39:21
2022368
Pranav Tanwar
2025-04-20 23:10:28


University Admin
Dashboard
University Dashboard
Sunday, April 20, 2025
Total Students
2
Registered Students
Roll No.
Name
Registration Date
101
AK
2025-04-20 22:39:21
2022368
Pranav Tanwar
2025-04-20 23:10:28


University Admin
Dashboard
University Dashboard
Sunday, April 20, 2025
Total Students
2
Registered Students
Roll No.
Name
Registration Date
101
AK
2025-04-20 22:39:21
2022368
Pranav Tanwar
2025-04-20 23:10:28


Grade Report
Signature Status: Valid
Name: Pranav Tanwar
Roll Number: 2022368
Date Issued: 2025-04-20 23:10:52
Course Code
Grade
CS101
80
ENG401
78
MATH201
63
PHYS301
70
Download PDF Report
Back to Dashboard
Back to Home