Name : Mariam Waleed

ID : 2205184

Name : Amr Khaled

ID : 2205220

Name : Mohamed Ahmed

ID : 2205249

Name : Shahd Mohamed

ID : 2205027

Name : Ahmed Elhefnawy

ID : 2205141



**Explaining the Concept of a Document Management System**

A document management system is an integrated web application that aims to provide a secure and organized environment for handling digital documents. Here is a detailed explanation of the concept

**Main Objective**

Providing a central platform for storing, managing, and sharing documents in a secure and efficient manner, with a focus on privacy and security.

**How the System Works**

Registration and Authentication:

The user creates an account using an email address and password.

Two-factor authentication can be enabled for increased security.

Google accounts can be used for quick registration.

Document Management:

Upload documents in multiple formats (PDF, DOCX, TXT)

Organize documents into folders

Search for documents using keywords

Security and Privacy:

Automatically encrypt documents upon storage

Digital sign documents to verify their integrity

Record all transactions in an activity log

Sharing and Collaboration:

Share documents with other users

Define access rights (read-only, edit)

Track changes to documents

Control Panel:

Easy-to-use interface for regular users

Administrator control panel for system management

Technologies Used

Backend: Python with Flask framework

Database: MySQL

Frontend: HTML, CSS, JavaScript

Security: Data encryption, two-factor authentication, digital signature

Additional Features:

Support for day and night modes

Customize user settings

Alerts and notifications for important activities

Usage reports and statistics The System

This system is ideal for companies and organizations that handle large volumes of documents and need to organize and protect them effectively.

**Code :**

Main application structure ( app.py) This function creates and initializes a Flask application, loading settings, initializing the database, and registering various blueprints.

User Model (models.py) The user model defines the structure of the user table in the database, and includes information such as email, encrypted password, role, and two-factor authentication settings.

Document model (models.py) The document model defines the structure of the document table, and includes information such as the file name, encrypted content, content signature, file type, file size, and upload information.

Document Management (controllers/documents.py) This section handles document uploading, validating the file, encrypting its content, and saving it to the database.

Two-factor authentication (utils/two\_factor.py) This class provides two-factor authentication functions, such as generating secret keys and verifying TOTP tokens.

Application settings (config.py) The settings file defines various application settings such as the secret key, database settings, and file upload settings.

Upload documents (controllers/documents.py) This section handles document uploads, verifies user permissions, and decrypts the content before sending it.

Auth system (controllers/auth.py) This section handles the login process, verifying user data and directing them to two-factor authentication if it is enabled.

Two-factor authentication verification (controllers/auth.py) This section deals with setting up two-factor authentication for the user, where it generates a secret key and a QR code.

Administrator Control Panel (controllers/admin.py) This section deals with the admin control panel, where it checks user permissions and provides user management functions.

File encryption and decryption (utils/security.py) These functions handle encryption and decryption of file content using the AES algorithm.

Signing documents number <|im\_start|> (controllers/documents.py) This section deals with signing documents with a private key.

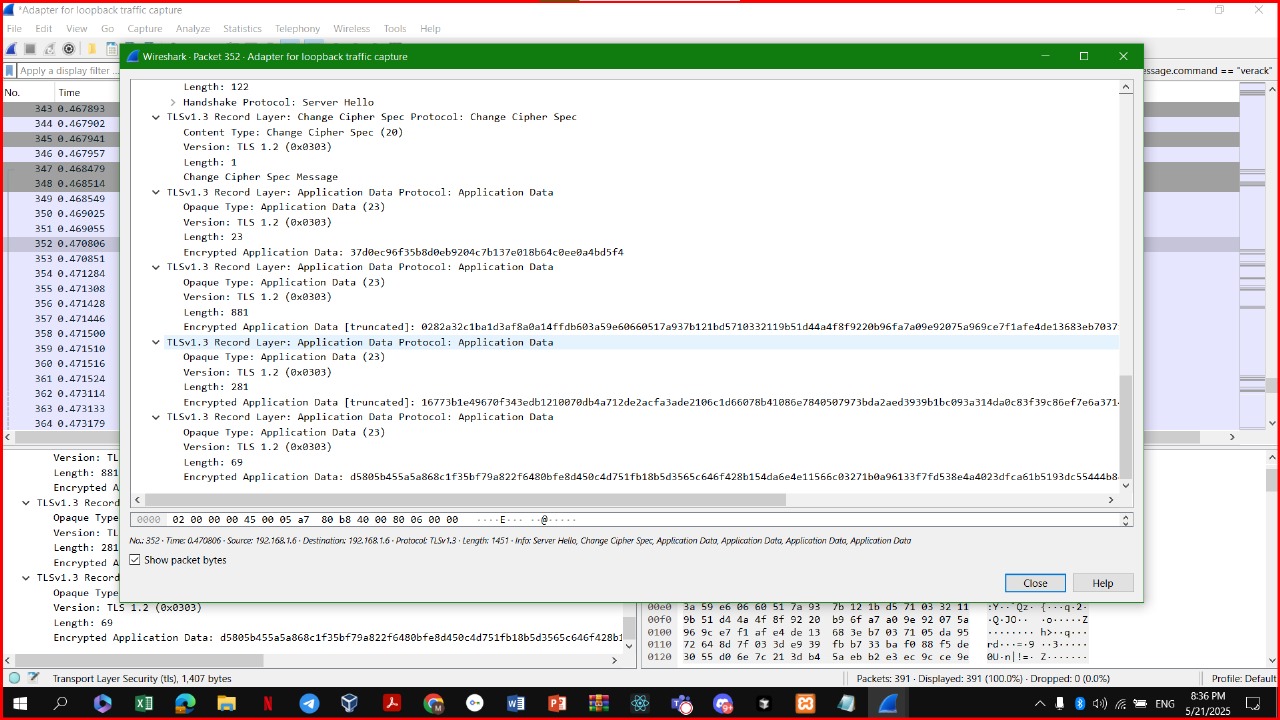
Digital Signature Verification (utils/security.py) This function verifies the digital signature of a document using the public key.

Event logging (controllers/documents.py) This function logs events to the audit log to track user activities.

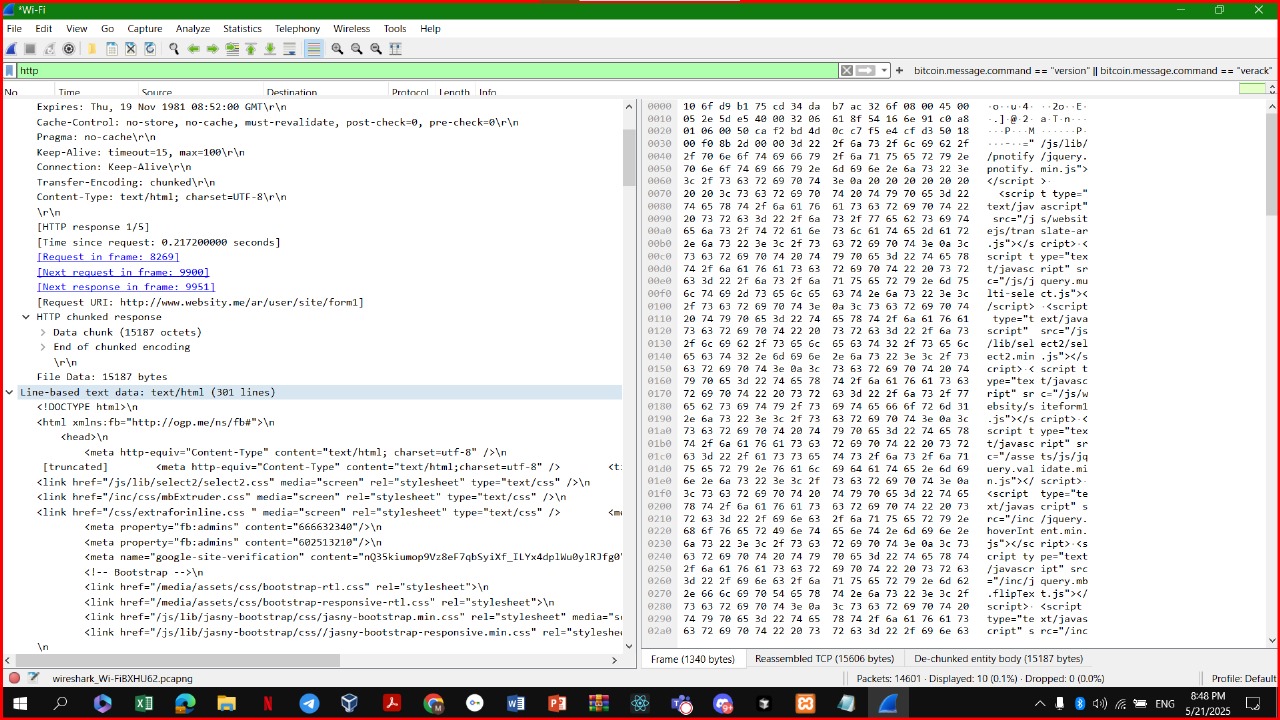
Create default profile picture ( create\_default\_profile.py ) This function creates a default profile picture for new users.

Form Validation (JavaScript) This function validates forms on the client side before sending them to the server.

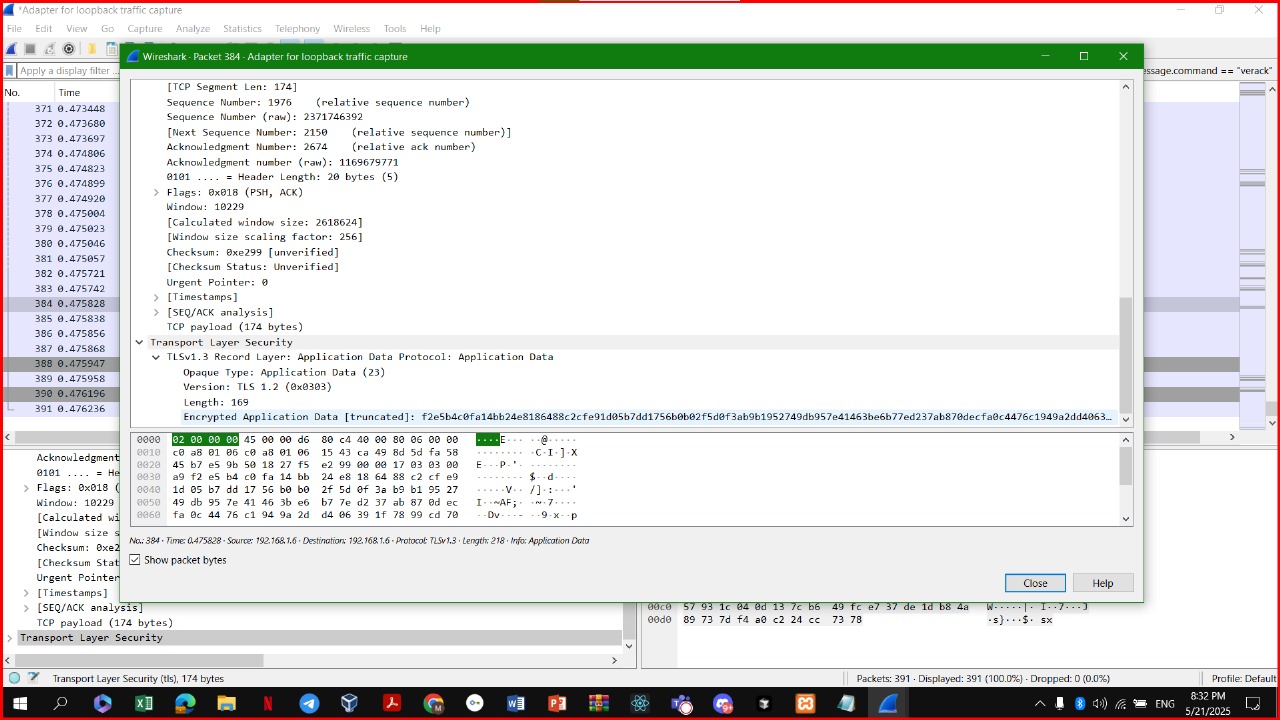
**Wireshark:**

Wireshark packet capture for loopback traffic, focusing on a TLSv1.3 handshake and subsequent application data. Key details include:

* **Time**: 0.470806 seconds
* **Handshake Protocol**: Server Hello, Change Cipher Spec
* **TLSv1.3 Record Layer**:
  + Change Cipher Spec (20 bytes)
  + Application Data (23 bytes, length: 23)
  + Encrypted Application Data (multiple instances, lengths: 23, 81, 281)
* **Version**: TLS 1.2 (0x0303)
* **Length**: Varies (23, 81, 281 bytes)
* **Encrypted Application Data**: Truncated hex values indicating secure data transfer.

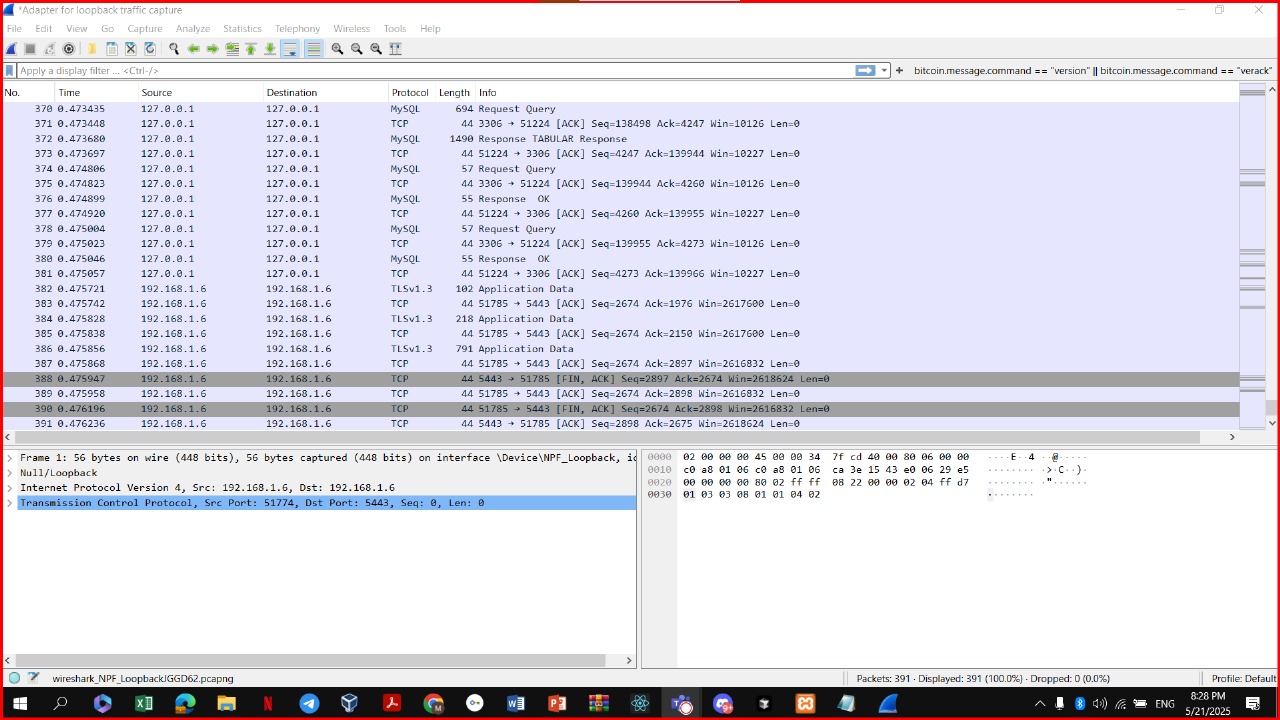
This Wireshark capture shows an HTTP response packet over Wi-Fi, with the following details:

* **Time**: Thu, 19 Nov 1981 08:52:06 GMT
* **Request URL**: <https://www.website.my/ar/user/site/form/>
* **Content-Type**: text/html; charset=UTF-8
* **Data Length**: 15187 bytes (text/html, 391 lines)
* **Response Details**:
  + HTML document with links to CSS (select2.css, stylesheet.css) and JavaScript (select2.js, bootstrap.min.js, jquery.js).
  + Meta properties for Google site verification and Bootstrap usage.
* **Data Size**: 15187 octets (de-chunked).

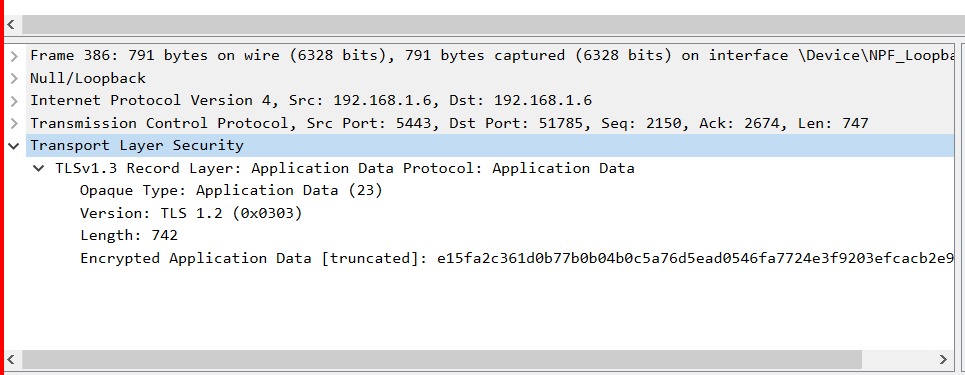


Wireshark packet capture for loopback traffic, specifically analyzing a TCP segment. The packet details include:

* **Time**: 0.473428 seconds
* **Sequence Number**: 174 (relative sequence number)
* **Next Sequence Number**: 2139 (relative sequence number)
* **Acknowledgment Number**: 2674 (relative acknowledgment number)
* **Header Length**: 20 bytes
* **Flags**: 0x018 (PSH, ACK)
* **Window Size**: 10229
* **Calculated Window Size**: 261
* **Checksum**: 0xe289 (unverified)
* **Urgent Pointer**: 0 (unverified)
* **TCP Payload**: 174 bytes

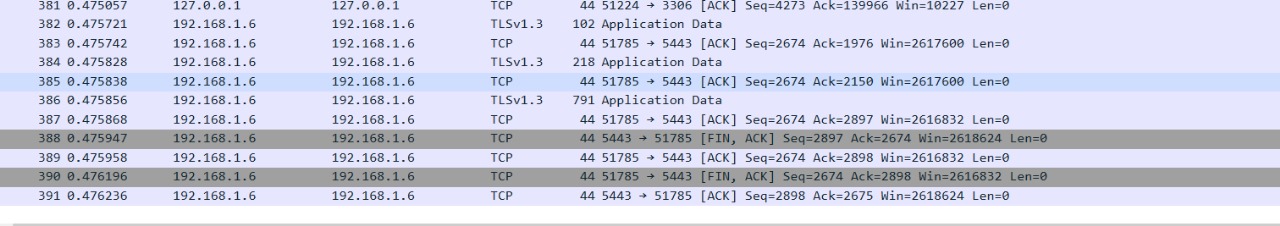
Wireshark capture shows loopback traffic involving MySQL and TCP/TLSv1.3 packets. Key details include:

* **MySQL Packets**:
  + Source/Destination: 127.0.0.1 (loopback)
  + Ports: 3306 (MySQL) and varying client ports (e.g., 51848, 54247)
  + Types: Response OK, Response TABULAR, Request Query
  + Lengths: Vary (e.g., 694 bytes, 14306 bytes)
* **TCP/TLSv1.3 Packets**:
  + Source/Destination: 192.168.1.6 (local network)
  + Ports: 51784, 5443
  + Protocol: TLSv1.3, Application Data
  + Lengths: Vary (e.g., 51785 bytes, 51783 bytes)
  + Flags: FIN, ACK; SEQ/ACK numbers provided
* **Frame Details**:
  + Frame 1: 56 bytes on wire, 56 bytes captured
  + Interface: \Device\NPF\_Loopback



Wireshark capture shows a TLSv1.3 packet with application data over loopback traffic. Key details include:

* **Frame**: 386, 791 bytes on wire (6328 bits), 791 bytes captured
* **Source/Destination**: 192.168.1.6 (Src) to 192.168.1.6 (Dst)
* **Ports**: 5443 (Src) to 51785 (Dst)
* **Sequence/Ack**: Seq: 2150, Ack: 2674, Len: 747
* **TLSv1.3 Record Layer**:
  + Protocol: Application Data
  + Opaque Type: Application Data (23)
  + Version: TLS 1.2 (0x0303)
  + Length: 742
  + Encrypted Application Data: Truncated hex value (e15fa2c361d0b77b0b04b0c5a76d5ead0546f7724e3f9203efcacb2e9)



This Wireshark capture shows loopback traffic involving TCP and TLSv1.3 packets. Key details include:

* **Source/Destination**: 192.168.1.6 to 192.168.1.6
* **Time**: Ranges from 0.475057 to 0.476236 seconds
* **Protocols and Details**:
  + **TLSv1.3**:
    - Packet 382: Application Data, Seq: 4273, Ack: 139966, Win: 10227, Len: 0
    - Packet 384: Application Data, Seq: 4273, Ack: 140970, Win: 2617600, Len: 0
    - Packet 386: Application Data, Seq: 4273, Ack: 141975, Win: 2617600, Len: 0
  + **TCP**:
    - Packet 383: Ack, Seq: 51785 + 5443, Ack: 1976, Win: 2617600, Len: 0
    - Packet 385: Ack, Seq: 51785 + 5443, Ack: 2897, Win: 261632, Len: 0
    - Packet 387: Ack, Seq: 51785 + 5443, Ack: 2898, Win: 261632, Len: 0
    - Packet 388: Fin, Ack, Seq: 2674, Ack: 2898, Win: 261632, Len: 0
    - Packet 389: Ack, Seq: 2898, Ack: 2675, Win: 261824, Len: 0