**CinemaHub Project Modules Overview**

This document explains the key modules used in the CinemaHub project, highlighting their functionality and how they contribute to the project.

**1. import getpass**

• Functionality:

The getpass.getpass() function is used to securely prompt users for a password without displaying the typed characters.

• Use Case:

In CinemaHub, we use getpass to securely handle user password inputs during the login and signup processes.

**2. from cryptography.fernet import Fernet**

• Functionality:

The Fernet class from the Cryptography library is used for symmetric encryption and decryption.

A single key is used to both encrypt and decrypt data.

• Use Case:

We use this module to securely encrypt user passwords, ensuring they are stored in an encrypted format for enhanced security within the CinemaHub platform.

**3. import re**

• Functionality:

The re module provides tools for pattern matching and text processing using Regular Expressions (RegEx).

• Use Case:

In CinemaHub, re is utilized to validate user input, such as ensuring proper email format during signup authentication.

**4. import uuid**

• Functionality:

The uuid module generates universally unique identifiers (UUIDs), which are 128-bit values that ensure uniqueness across systems.

• Use Case:

This module is used to generate unique IDs for booking records, ensuring no conflicts between different bookings.

**5. import os**

• Functionality:

The os module allows interaction with the operating system, including file and directory management, environment variable access, and process handling.

• Use Case:

CinemaHub uses os to manage files such as logs, configuration files, and PDFs, ensuring platform-independent operations.

**6. from fpdf import FPDF**

• Functionality:

The FPDF module is used to generate PDF documents programmatically, including text, images, and tables.

• Use Case:

We use FPDF to generate and display CinemaHub tickets within PDF documents.

**7. from tabulate import tabulate**

• Functionality:

The tabulate module formats data into neat, readable tables.

• Use Case:

It is used to display CinemaHub information, such as theater schedules, showtimes, and seat availability in a tabular format.

**8. from datetime import datetime**

• Functionality:

The datetime module provides classes for manipulating dates and times.

• Use Case:

We use it for printing the current date and saving files with date-based filenames, ensuring unique file naming.

**9. from PyPDF2 import PdfReader, PdfWriter**

• Functionality:

PdfReader is used for extracting content from PDF files, and PdfWriter is used to create or modify PDFs.

• Use Case:

These are used for reading, manipulating, and merging PDF documents within CinemaHub, such as extracting and modifying ticket details.

**10. from reportlab.pdfgen import canvas**

• Functionality:

The canvas module allows for drawing shapes, text, and images within PDFs.

• Use Case:

In CinemaHub, canvas is used to create custom PDF templates for tickets, including personalized information such as movie names and showtimes.

**11. from io import BytesIO**

• Functionality:

BytesIO provides an in-memory buffer to read and write bytes, simulating file operations without saving to disk.

• Use Case:

This is particularly useful for handling PDF files entirely in memory, allowing CinemaHub to generate and serve tickets dynamically without relying on file storage.

**12. from reportlab.pdfbase import pdfmetrics**

• Functionality:

pdfmetrics is used for managing font metrics and ensuring correct text rendering in PDFs.

• Use Case:

We use it to manage font properties, ensuring accurate and styled text rendering in CinemaHub’s generated PDFs.

**13. from reportlab.pdfbase.ttfonts import TTFont**

• Functionality:

TTFont allows the use of TrueType fonts (TTF) in PDFs, enabling custom fonts for personalized content.

• Use Case:

CinemaHub utilizes this to embed custom fonts into tickets for a more polished and professional appearance.

**14. import mysql.connector**

• Functionality:

mysql.connector provides a Python interface to connect to MySQL databases, enabling interaction with data such as executing SQL queries and handling transactions.

• Use Case:

CinemaHub uses this module to connect to its MySQL database, store user and booking data, and perform database operations like inserting, updating, and deleting records.