Research Article

Hyperhidrosis – An Interactive Web Application for Detection and Accurate Diagnosis of Hyperhidrosis

Nishna Sarkar[1], Pujaa B[2]*

1. Stella Maris College, Chennai, Tamilnadu, India.
2. Assistant Professor, Stella Maris College, Chennai, Tamilnadu, India.

***Corresponding author's E-mail:** pujaa@stellamariscollege.edu.in

Received: 03-09-2024; **Revised:** 20-01-2025; **Accepted:** 08-02-2025; **Published online:** 20-02-2025.

ABSTRACT

Hyperhidrosis, characterized by excessive sweating, significantly impacts individuals' quality of life. To address this, we developed "Hyperhidrosis," a web application designed to support and empower those affected by the condition. Built using Flask, VSCode, WampServer, Bootstrap, and integrated with the PubChem API, the application offers comprehensive resources and tools for managing hyperhidrosis. During backend development, Flask and VSCode were utilized to implement critical functionalities such as user authentication, database management, and external API integration. Flask's modular architecture ensured seamless communication between frontend and backend components, facilitating smooth user interactions and efficient data exchange. For frontend development, Bootstrap was used to create a responsive and visually appealing user interface, enhancing user experience. The integration of the PubChem API provided real-time access to chemical information, significantly enriching the application's value for users seeking treatment options and related resources. Comprehensive testing validated the application's functionality, performance, and compatibility across various devices and browsers. Performance evaluations revealed acceptable response times, supported by efficient handling of PubChem API requests and optimization measures. By prioritizing performance, reliability, and user-centric design, "Hyperhidrosis" aims to be a trusted resource for the hyperhidrosis community, contributing to improved health outcomes and quality of life for affected individuals.

Keywords: Hyperhidrosis, Web application, Flask, PubChem API.

INTRODUCTION

Medical professionals generally have a positive view of e-health, especially those familiar with technology, though they seek adequate resources, technology, and training for effective implementation[1]. Specific web applications for hyperhidrosis (HH) are limited, but international platforms like the International Hyperhidrosis Society (IHhS) and Hyperhidrosis UK offer information, support, and advocacy, promoting awareness and improving treatment options[2]. A user-centered design approach was employed to understand the target audience's demographics, behaviors, and pain points through thorough user research, literature, and usability testing. This informed the design decisions throughout development. Python was used for backend functionalities, with VSCode providing a collaborative environment for code iteration based on user feedback. WampServer enabled rapid prototyping and user testing by hosting the application locally, allowing for quick iterations and refinements. Bootstrap ensured a visually appealing and responsive user interface for various devices, enhancing usability and engagement[3]. The integration of PubChem into the development workflow enhanced the web application by enabling the retrieval of pharmaceutical compound information. Python's versatility allowed easy interaction with PubChem's RESTful API to retrieve data such as chemical identifiers, molecular structures, and pharmacological properties. Visual Studio Code provided an efficient environment for writing scripts to query and process PubChem data[4]. WampServer facilitated local hosting and seamless integration of these functionalities. Bootstrap's responsive components displayed the retrieved information in a user-friendly manner, ensuring accessibility across devices. This integration empowered users to retrieve detailed information about medicines and chemical compounds, enhancing the application's capabilities[5].

MATERIALS AND METHODS

The main workflow of the study is illustrated below (Fig.1).

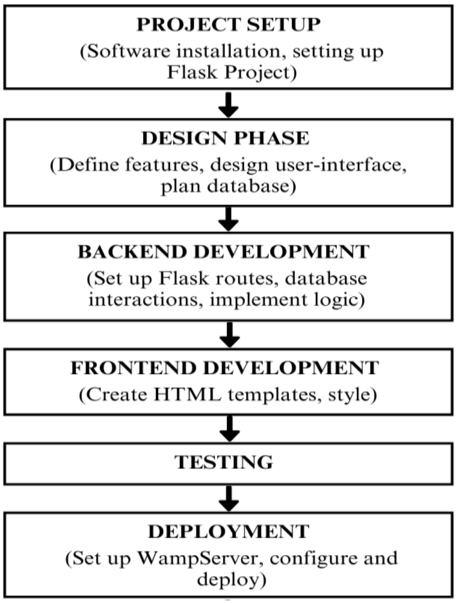

PROJECT SETUP
(Software installation, setting up Flask Project)
DESIGN PHASE
(Define features, design user-interface, plan database)
BACKEND DEVELOPMENT
(Set up Flask routes, database interactions, implement logic)
FRONTEND DEVELOPMENT
(Create HTML templates, style)
TESTING
DEPLOYMENT
(Set up WampServer, configure and deploy)

Figure 1: Workflow

Visual Studio Code (VSCode) as IDE

Visual Studio[6] is a cross-platform editor available for Windows, macOS, and Linux[7]. It features a robust ecosystem of extensions supporting various languages and runtimes, including C++, C#, Java, Python, PHP, Go, and .NET, with built-in support for TypeScript, JavaScript, and Node.js[8]. VSCode includes built-in version control integration and debugging capabilities, allowing efficient issue troubleshooting. Its extensive extension ecosystem enables customization to suit specific needs. The Python extension is particularly useful for Python development. This platform was extensively used for the web application's development to ensure smooth functioning and user information storage (The VSCode Team).

Python-Flask as Web Application Framework

Python-Flask is a lightweight and versatile web development framework known for its simplicity, flexibility, and ease of use. It is ideal for developers who prefer a minimalistic approach, as it does not impose specific project structures or dependencies. Flask provides essential features like routing, templating, and request handling, and can be extended through third-party extensions for database integration, authentication, and form handling[9]. Its lightweight nature and modular design make it suitable for both simple prototypes and complex web applications, allowing developers to build, customize, and scale their projects efficiently[10].

Bootstrap, CSS, and JavaScript for Frontend Development

Visually appealing and interactive web applications can be built using Bootstrap as it provides the required pre-designed components such as JavaScript and CSS. JavaScripts are known for their versatility in allowing the user input to interact with the web APIs CSS helps in customizing the visual layout of the web pages.

WampServer as Local Development Environment

WampServer provides a comprehensive and user-friendly local development environment, including Apache, MySQL, and PHP, for creating web applications easily. Apache serves as the local web server, handling HTTP requests. MySQL is the local database server for managing data, and PHP enables server-side scripting for dynamic web applications. WampServer, available for both 32- and 64-bit systems under the GPML license, allows for server tuning without configuration file changes[11].

PubChem as a Medicine Database / Repository

PubChem[12] is a chemical information resource at the U.S. National Center for Biotechnology Information (NCBI), established in 2004. It serves the cheminformatics, chemical biology, medicinal chemistry, and drug development communities. PubChem's data is organized into three interconnected databases: Substance (SID), Compound (CID), and BioAssay (AID). The Substance database contains sample descriptions from depositors, the Compound database has unique chemical structures, and the BioAssay database includes bioactivity screens and biological test results from various contributors. The PubChem REST API, provided by NCBI, allows programmatic access to PubChem data. It offers endpoints for tasks like searching chemical compounds, retrieving detailed compound information, accessing bioactivity data, and performing structure similarity searches. The API supports data formats like JSON, XML, and SDF, enabling flexible integration into applications[13].

Installation of Software (VSCode, Python, Flask, WampServer)

To install Visual Studio Code (VSCode) on Windows 10, the official VSCode website was visited, and the download was initiated by clicking the download button. After downloading the "VSCodeSetup.exe" file, it was run, and the on-screen instructions were followed to complete the installation. VSCode was then launched from the Start menu or a desktop shortcut. For Python in VSCode, the Extensions view was opened by clicking the square icon in the sidebar or pressing Ctrl+Shift+X. The "Python" extension provided by Microsoft was searched and installed. VSCode was reloaded to activate the extension, enabling Python features like syntax highlighting, linting, and debugging. Separately, Python was installed by downloading the appropriate installer from the official Python website. The installer was executed, and the on-screen instructions were followed, including choosing the installation directory and adding Python to the system PATH. After installation, Python was accessible via the command line or by running Python scripts. Additional packages could be installed using pip. To install WampServer, the installer package was downloaded from the official WampServer website. The installation process involved selecting the installation directory, accepting the license agreement, and choosing components to install (Apache, MySQL, PHP). Configuration settings for Apache and MySQL ports were set up during the installation. Once completed, WampServer was launched.

Setting up Flask Project

To set up a Flask project in VSCode, a new folder was created, and the terminal was opened to this folder. Flask was installed using the command `pip install flask`. A new Python file, typically named "app.py," was created, and the Flask application code was written in it. The code included the creation of a Flask application object with `app = Flask(__name__)`, and modules like `render_template`, `request`, `redirect`, and `session` were imported for handling HTML rendering, incoming request data, client redirection, and user-specific session information. MySQLdb was used for database interaction, connecting to the WampServer database to securely store data[14].

Design Phase

The design phase involved planning the web application interface, creating HTML files for "user," "register," "login," and "logout" pages to manage user information. HTML files were also created to handle symptoms, treatments related to hyperhidrosis (HH), and displaying compound structures retrieved from PubChem. The database was created using MySQL in the WampServer environment, with tables for each HTML file to store and retrieve values securely.

Backend Development

During backend development, Flask routes were established to map URLs to Python functions for navigation and functionality. Database interactions were implemented using `flask_mysqldb`, facilitating communication between the Flask application and the MySQL database. Custom business logic was integrated to address specific application needs.

Configuration variables included `app.secret_key` for session security and MySQL settings:

- `MYSQL_HOST` set to 'localhost'

- `MYSQL_USER` set to 'root'

- `MYSQL_PASSWORD` for user authentication

- `MYSQL_DB` set to 'user-system' indicating the database used by the application.

Frontend Development

The user account dashboard was styled using Bootstrap, featuring a container with form groups displaying the user's name and a logout link, dynamically retrieved from session data using Flask. A link to the symptoms page facilitated navigation, and the layout was made responsive with Bootstrap's grid system. The user login form, styled with Bootstrap, included input fields for email and password, a login button, and a warning message dynamically rendered with Flask. A link to the registration page was provided for new users. Similarly, the user registration form featured input fields for name, email, password, age, gender, and occupation, with dynamic warning messages and a link to the login page. A treatment form for hyperhidrosis was also created with Bootstrap, allowing users to indicate if they had sought treatment and select from various treatment options. Conditional JavaScript displayed additional components based on user input, and form data was submitted to the 'treat' Flask route using the POST method. A compound information page was styled with Bootstrap, dynamically rendering compound data from the server using Flask's template engine. It displayed details like the compound ID, name, molecular formula, and a 2D structure image, with conditional rendering for missing information.

RESULTS AND DISCUSSION

The hypothesis for developing a web application on hyperhidrosis (HH) focuses on assessing the effectiveness of digital platforms in providing information and support for individuals affected by HH. This section presents the results of our web application development project, highlighting the implementation and performance of the application built using Flask, VSCode, WampServer, Bootstrap, and PubChem. We discuss the outcomes of the backend and frontend development phases, integration, and testing results. The developed web application, "Hyperhidrosis," aims to provide information and resources for individuals dealing with HH, a medical condition characterized by excessive sweating. The user interface, leveraging Bootstrap's responsive design capabilities, is intuitive and accessible, featuring educational content and treatment resources. In the backend development phase, Flask and VSCode were used to implement core functionalities. Flask's lightweight framework enabled efficient routing, template rendering, and database management. The integration of the PubChem API enriched the application with real-time chemical information retrieval capabilities. User authentication and authorization were securely handled using Flask-Login. Despite challenges such as API rate limiting and data parsing complexities, effective solutions were devised. Future efforts will prioritize optimizing database queries and further API integrations for enhanced functionality and scalability. The frontend development phase, utilizing Bootstrap and VSCode, resulted in visually appealing and user-friendly interfaces. Responsive layouts were created using Bootstrap's components. The integration of frontend and backend components was facilitated by Flask's modular architecture. Comprehensive testing validated functionality, performance, and compatibility across devices. PubChem API integration enhanced the application with real-time chemical information retrieval. Performance evaluations showed acceptable response times, even under moderate user loads, with efficient handling of PubChem API requests. The homepage (Fig.2) focuses on a user-centered experience with "Registration" for new users and "Login" for registered users (Fig.3). An "About" button provides information about the web application's aim to address HH, raising awareness about its symptoms and treatment options, and offering chemical information about available medications (Fig.4 and 5). The results of our web application development project demonstrate the successful implementation of Flask, VSCode, WampServer, and Bootstrap in creating a functional and user-friendly application for individuals dealing with hyperhidrosis (Fig.6 and 7). While the application meets the initial objectives, there are opportunities for further refinement and enhancement, particularly in terms of performance optimization and user engagement features.

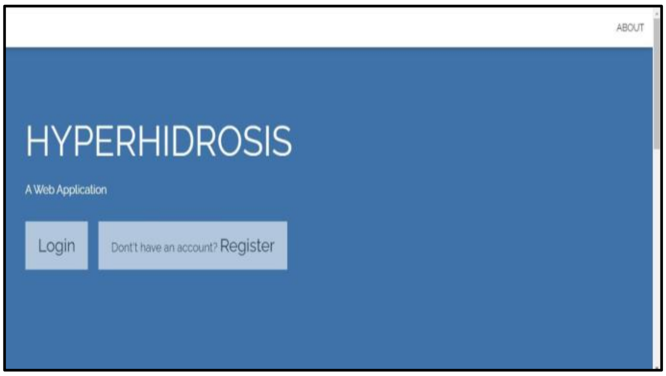
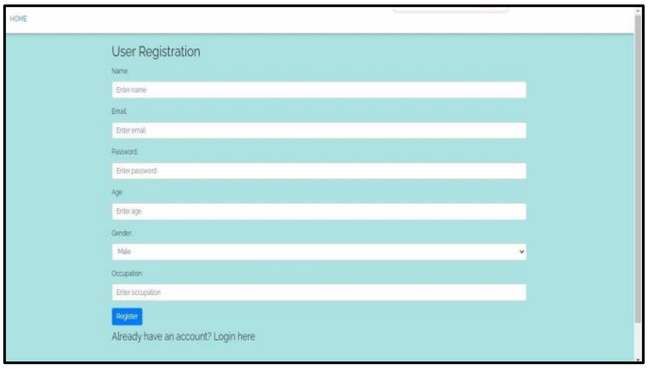


Figure 2: Homepage of the web application



User Registration

Name:
Enter name

Email:
Enter email

Password:
Enter password

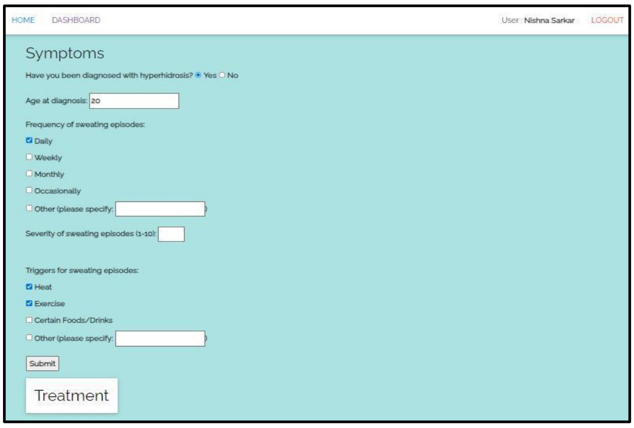
Age:
Enter age

Gender:
Male

Occupation:
Enter occupation

Already have an account? Login here

Figure 3: Display of "Registration" page



User : Nishna Sarkar

Symptoms

Have you been diagnosed with hyperhidrosis? ● Yes ○ No

Age at diagnosis: 20

Frequency of sweating episodes:

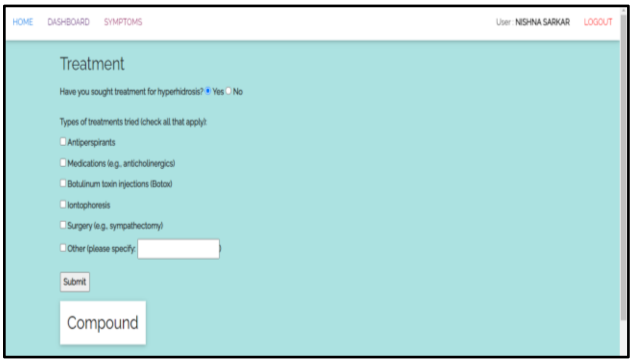
☑ Daily
☐ Weekly
☐ Monthly
☐ Occasionally
☐ Other (please specify:)

Severity of sweating episodes (1-10):

Triggers for sweating episodes:

☑ Heat
☑ Exercise
☐ Certain Foods/Drinks
☐ Other (please specify:)

Figure 4: Full display of "Symptoms" page with the questionnaire



User : NISHNA SARKAR

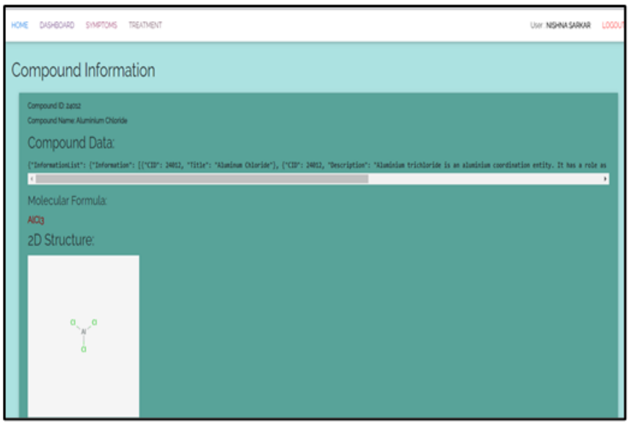
Treatment

Have you sought treatment for hyperhidrosis? ● Yes ○ No

Types of treatments tried (check all that apply):

☐ Antiperspirants
☐ Medications (e.g., anticholinergics)
☐ Botulinum toxin injections (Botox)
☐ Iontophoresis
☐ Surgery (e.g., sympathectomy)
☐ Other (please specify:)

Figure 5: Display of "Treatment" page



User : NISHNA SARKAR

Compound Information

Compound ID: 24012
Compound Name: Aluminium Chloride

Compound Data:

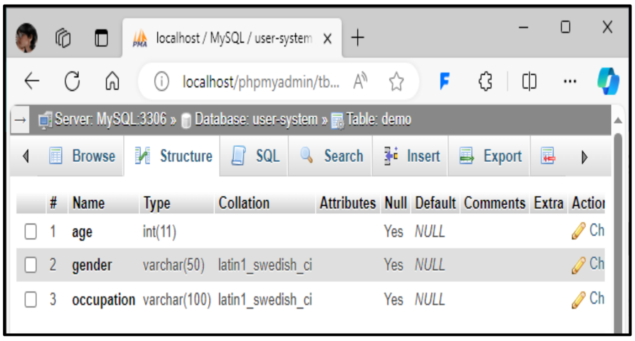
```
{"InformationList": {"Information": [{"CID": 24012, "Title": "Aluminum Chloride"}, {"CID": 24012, "Description": "Aluminium trichloride is an aluminium coordination entity. It has a role as
```

Molecular Formula:

AlCl3

2D Structure:

Figure 6: Display of "Compound Information" page



Server: MySQL:3306 » Database: user-system » Table: demo

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Actio
☐	1	age	int(11)			Yes	NULL			Ch
☐	2	gender	varchar(50)	latin1_swedish_ci		Yes	NULL			Ch
☐	3	occupation	varchar(100)	latin1_swedish_ci		Yes	NULL			Ch

Figure 7: The "User" details stored in database made in "phpMyAdmin"

Hyperhidrosis (HH) is not widely recognized in India despite growing awareness in other parts of Asia[15]. Medical literature on HH covers treatments such as topical antiperspirants, oral medications, iontophoresis, botulinum toxin injections (Botox), and surgical methods, discussing their effectiveness, tolerability, and long-term outcomes[16]. Complementary therapies like acupuncture and herbal supplements have also been explored for symptom relief[17]. Patient experiences are captured through qualitative research and patient-reported outcome measures[18]. The development of a web application using Python, Visual Studio Code, WampServer, and Bootstrap integrated with PubChem aims to create a user-centered platform for HH management. Python handles backend logic and PubChem integration, Visual Studio Code supports collaborative development, WampServer facilitates local hosting, and Bootstrap ensures a responsive user interface. This approach addresses the needs of individuals with HH by providing medication insights and information.

During development, challenges such as configuring Flask routes, handling database interactions, and integrating external APIs like PubChem were encountered and resolved through iterative development cycles. The application empowers HH sufferers by providing accessible information and resources, and the PubChem integration offers access to scientific knowledge that could advance HH treatment. Future improvements could include personalized content recommendations, community engagement tools, and data visualization capabilities. Ongoing user feedback will guide the application's evolution. Future research might explore the use of emerging technologies, like artificial intelligence and machine learning, to create more personalized and effective web-based interventions for HH. Collaboration among researchers, healthcare professionals, and technology experts is essential for developing innovative solutions to improve patient outcomes and the healthcare experience for those with HH.

CONCLUSION

Understanding the causes and challenges of hyperhidrosis (HH) is essential, and the "Hyperhidrosis" web application aims to address these needs by utilizing Python-Flask for backend development and Bootstrap for frontend design. Integration with the PubChem API enriches the platform with real-time chemical information retrieval, providing a user-friendly experience for individuals seeking resources and support for HH.Future research could explore leveraging technologies like AI and machine learning to create more personalized web-based interventions. Potential features such as symptom tracking, treatment reminders, and online forums, along with enhanced PubChem API integration, can continually improve the user experience. Scalability measures like database indexing and server optimization will be implemented to handle increasing traffic efficiently. In conclusion, the development of the "Hyperhidrosis" web application using Python-Flask, VSCode, WampServer, and Bootstrap represents a significant achievement in providing valuable resources and support for individuals affected by HH. The successful implementation of these technologies underscores their effectiveness in creating robust and scalable web applications. Moving forward, efforts will focus on improving performance, enhancing user engagement, and expanding features to better serve users.

Source of Support: The author(s) received no financial support for the research, authorship, and/or publication of this article

Conflict of Interest: The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

REFERENCES

1. Ruiz Morilla MD, Sans M, Casasa A, Giménez N. Implementing technology in healthcare: insights from physicians. BMC Medical Informatics and Decision Making. 2017 Dec 27;17(1):92. Doi: 10.1186/s12911-017-0489-2.
2. Doolittle J, Walker P, Mills T, Thurston J. Hyperhidrosis: an update on prevalence and severity in the United States. Archives of Dermatological Research. 2016 Dec 15;308(10):743–9. Doi: 10.1007/s00403-016-1697-9.
3. Chiaretta S. Front-end Development with ASP.NET Core, Angular, and Bootstrap. Wiley; 2018. Doi: 10.1002/9781119549321.
4. Amann S, Proksch S, Nadi S, Mezini M. A Study of Visual Studio Usage in Practice. In: 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER). IEEE; 2016. p. 124–34. Doi: 10.1109/SANER.2016.39.
5. bin Uzayr S. Getting Started with Python Programs in Visual Studio Code. In: Optimizing Visual Studio Code for Python Development. Berkeley, CA: Apress; 2021. p. 47–91. Doi: 10.1007/978-1-4842-7344-9_2.
6. Visual Studio Code - Code Editing. Redefined. [Internet] 2021; Available from: https://code.visualstudio.com/
7. Introducing Visual Studio Code. In: Visual Studio Code. Wiley; 2019. p. 1–12. Doi: 10.1002/9781119588238.ch1.
8. Rask JK, Madsen FP, Battle N, Macedo HD, Larsen PG. Visual Studio Code VDM Support. In Fitzgerald J, Oda T, Macedo HD, editors, Proceedings of the 18th International Overture Workshop. 2021. p. 35-50.
9. Salo K. Comparative study on Python web frameworks: Flask and Django. 2023.
10. Pereira JD, Silva R, Antunes N, Silva JLM, de França B, Moraes R, et al. A platform to enable self-adaptive cloud applications using trustworthiness properties. In: Proceedings of the IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems. New York, NY, USA: ACM; 2020. p. 71–7. Doi: 10.1145/3387939.3391608.
11. Ipswich D. Setting up a WAMP server on your windows desktop. *Technology Now at Smashwords*. 2011.
12. Pubchem. National Library of Medicine - National Center for Biotechnology Information. Available from: https://pubchem.ncbi.nlm.nih.gov/
13. Wang Y, Xiao J, Suzek TO, Zhang J, Wang J, Bryant SH. PubChem: a public information system for analyzing bioactivities of small molecules. Nucleic Acids Research. 2009 Jul 1;37(Web Server):W623–33. Doi: 10.1093/nar/gkp456.
14. Wampserver – Installers, Updates, Addons, Tools. Available at: https://wampserver.aviatechno.net/
15. Muthusamy A. A Study on the Impact of Hyperhidrosis on the Quality of Life among College Students. Journal of Clinical and Diagnostic Research. 2016; Doi: 10.7860/JCDR/2016/19495.8061.
16. Stolman LP. Treatment of Hyperhidrosis. Dermatologic Clinics. 1998 Oct;16(4):863–9. Doi: 10.1016/S0733-8635(05)70062-0.
17. Cayir Y, Engin Y. Acupuncture for Primary Hyperhidrosis: Case Series. Acupuncture in Medicine. 2013 Sep 1;31(3):325–6. Doi: 10.1136/acupmed-2013-010391.
18. Wade R, Jones-Diette J, Wright K, Layton AM, Woolacott N. Hyperhidrosis quality of life measures: review and patient perspective. Journal of Dermatological Treatment. 2019 Apr 3;30(3):303–8. Doi: 10.1080/09546634.2018.1506080.