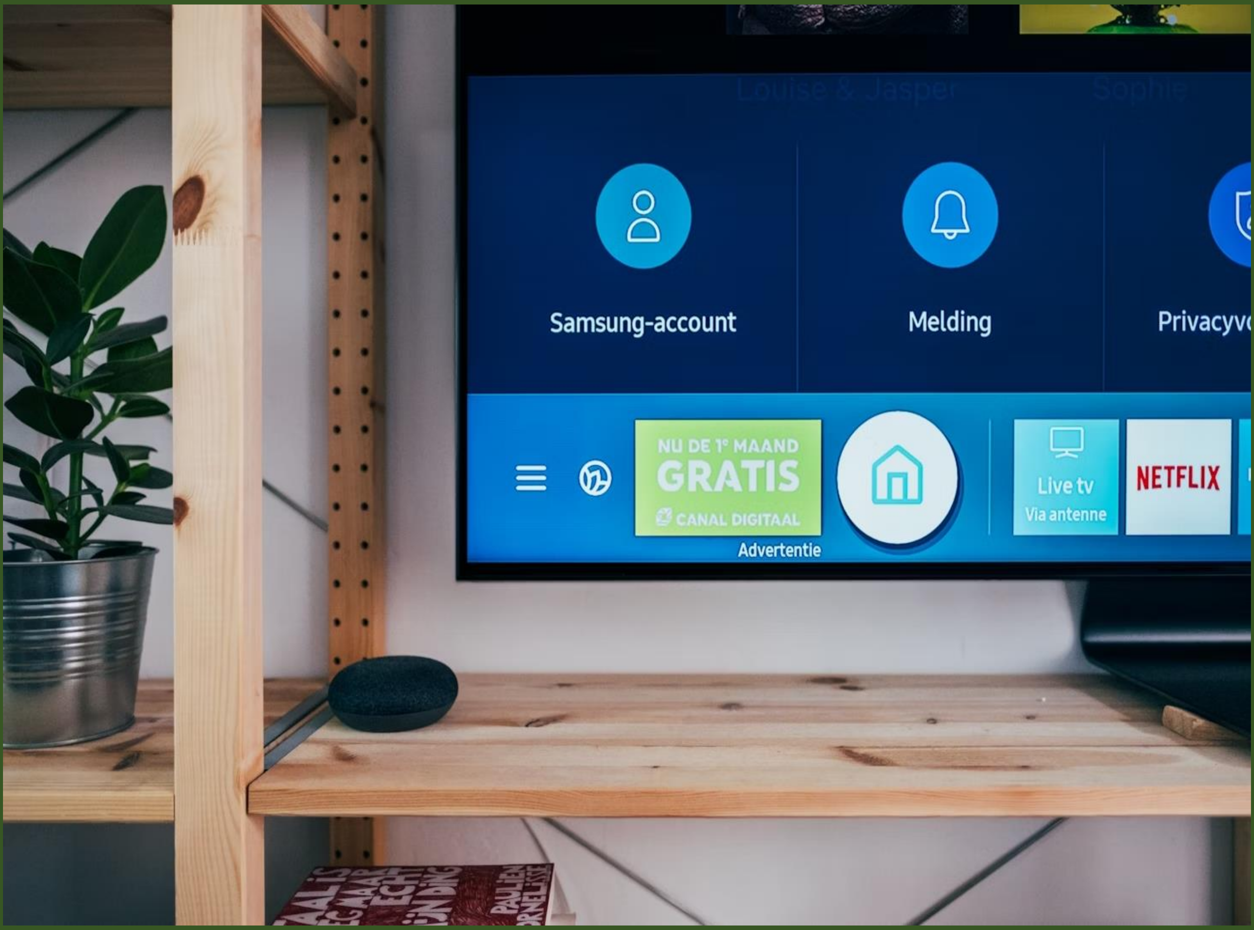
CONTEST-AZM

تونشانِ عزمِ عالیشان



STREAM EASE WEB APPLICATION

Aptech LEARNING

Unleash your potential

Table of Contents

1. Background and Necessity for the Website

Streaming services have become increasingly popular and have transformed the way we consume media. Streaming is an immediate and continuous method of accessing content from the Internet. It has become the predominant way for people to experience music and videos. With the increasing number of streaming services available, users often subscribe to multiple platforms. Keeping track of various subscriptions, managing billing cycles, and organizing content across different services can become cumbersome. A Web Application can offer a centralized platform where users can manage and track their subscriptions, view upcoming payments, and maintain a comprehensive overview of their streaming services in one place.

2. Proposed Solution

The proposed solution is a Website called 'StreamEase' that allows users to track and manage their streaming services and subscriptions in one centralized platform. The application should provide features such as adding/modifying/deleting streaming providers, marking favorite shows, setting reminders for upcoming streaming payments, reminders for upcoming shows, searching, sorting, and filtering functionalities.

3. Purpose of the Document

The proposed solution is a Web Application called 'StreamTrace' that allows users to track and manage their streaming services and subscriptions in one centralized platform. The application should provide features such as adding/modifying/deleting streaming providers, marking favorite shows, setting reminders for upcoming streaming payments, reminders for upcoming shows, searching, sorting, and filtering functionalities.

4. Scope of Project

This Web application will be a responsive and visually appealing one to be used by individuals. The scope of StreamTrace Web application can encompass various functionalities and features. It will be designed to aid users in maintaining various subscriptions and related information as subscription renewal date, subscription payments, favorite shows, and so on. However, the application will not have any feature/functionality for implementing or authenticating payment or content streaming. These actions are beyond the scope of this application.

5. Constraints

Streaming services typically handle a large number of concurrent users who expect high-quality video playback and seamless streaming experiences. Ensuring scalability and performance constraints involves optimizing the infrastructure to handle increased traffic, reducing buffering times, and delivering a consistent streaming experience across different devices and network conditions. These features are beyond the scope of this Web application.

6. Functional Requirements

1. Home Page

- Dashboard: Displays previously watched shows and subscription details.
- Introduction: Welcoming message and brief overview of StreamTrace.

2. Account Registration

- Fields Required: Name, Email ID, Contact Number, Username, and Password.

- Validation:
- Email ID: Must be in the correct format (e.g., user@example.com).
- Other Fields: Appropriate error-checking for data accuracy and completeness.
- Client-Side Validation: Ensure real-time feedback on form entries.

3. Login

- Access: Allows registered users to log in and access StreamTrace features.
- Profile Information: Includes details such as age and other personal information.

4. Settings

- Profile Management: Options to Create, Update, and Delete profiles.
- Address Management: Add and manage addresses associated with the user’s profile.

5. Streaming Service Provider Management

- Manage Providers: Add, update, and delete service providers.
- Details: Include Name, Logo, and other relevant information.

6. Subscription Management

- Manage Subscriptions: Add, edit, or cancel subscriptions.
- Subscription Details: Include Name, Logo, Subscription URL, Renewal Date, Amount, etc.
- Reminders: Notifications for upcoming payments or renewal dates.

7. Recommendations

- Personalized Content: Based on user preferences, viewing history, and ratings.

8. Watchlist and Favorites Management

- Watchlist: Create and manage a list of movies, TV shows, or music albums.
- Favorites: Mark and track favorite content.
- Watched Content: Track content that has already been watched.

9. Search/Sort/Filter

- Search: Find specific shows, movies, etc.
- Sort: Organize content based on various criteria.
- Filter: Narrow down content based on user preferences.

10. Contact Us

- Details Displayed: Email ID, Address, and Contact Number of the organization.

11. Feedback

- Feedback Form: Allow users to submit their feedback about the application.

12. Sitemap

- Sitemap: Provide an overview of the application flow. Add to the home page for easy navigation.

7. Non-Functional Requirements

There are several non-functional requirements that should be fulfilled by the Website. The Website should be:

- Safe to use: The Website should not result in any malicious downloads or unnecessary file downloads.
- Accessible: The Website should have clear and legible fonts, user-interface elements, and navigation elements.
- User-friendly: The Website should be easy to navigate with clear items and other elements and easy to understand.

- Operable: The Website should operate in a reliably efficient manner.
- Performance: The Website should demonstrate high value of performance through speed and throughput. In simple terms, the Website should be fast to load and page redirection should be smooth.
- Capacity: The Website should support a large number of users.
- Availability: The Website should be available 24/7 with minimum downtime.
- Compatibility: The Website should be compatible with the latest browsers.

These are the bare minimum expectations from the project. It is a must to implement the functional and non-functional requirements given in this SRS. Once they are complete, you can use your own creativity and imagination to add more features if required.

8. Interface Requirements

8.1 Hardware

- Intel Core i5 Processor or higher
- 8 GB RAM or higher
- Color SVGA
- 500 GB Hard Disk space

- Mouse
- Keyboard

8.2 Software

Technologies to be used:

1. Frontend:

- HTML5: For structuring the content on the web pages.
- CSS3: For styling and designing the web pages.
- Bootstrap (optional): For responsive design and UI components.
- JavaScript: For client-side scripting and dynamic content.
- Figma Toolkit: For designing UI/UX layouts and prototypes.
- jQuery: For simplified HTML document traversal and manipulation.
- AngularJS/Angular 9/ReactJS: For building interactive and dynamic user interfaces.
- ReactJS: For creating reusable UI components and managing application state.
- XML: For data exchange and configuration.

2. Client and Server:

- Java:
    1. Version: 9 or higher.
    2. Frameworks: Java EE 7 or higher / Jakarta EE 9 or higher.
    3. IDE: Apache NetBeans IDE / Eclipse (latest version).
    4. Servers: Apache Tomcat 10.0 or higher, GlassFish 6.0 or higher.
    5. Libraries: Relevant libraries for Java EE/Jakarta EE.
- C#:
    1. Version: 7.0 or higher.
    2. IDE: Visual Studio IDE 2019 or higher.
    3. Frameworks: ASP.NET MVC and Core.
    4. Libraries: Related libraries for ASP.NET.
- PHP:
    1. Version: 7.0 or higher.
    2. Frameworks: Laravel Framework.
    3. Hosting (optional): Homestead.
- Python:
    1. Version: 3.0 or higher.
    2. IDE: PyCharm IDE.
    3. Frameworks: Django 4.0.2 or higher / Flask.
- Node.js and Express:

1. Node.js: JavaScript runtime for building scalable network applications.
2. Express: Web application framework for Node.js to handle routing and middleware.

3. For Hosting (optional):

1. XAMPP: Latest version for local development and hosting.

4. Data Store:

2. MySQL: 5.7 or higher.
3. SQL Server: 2016 or higher.

9. Project Deliverables

You will design and build the project and submit it along with a complete project report that includes:

- Problem Definition
- Design Specifications
- Diagrams:
- Flowcharts for various activities
- Data Flow Diagrams (DFDs)
- Other relevant diagrams
- Database Design:
- Database scripts to be provided
- Source Code
- Installation Guide

- User Manual along with test data and login credentials

Documentation is considered as a very important part of the project. Ensure that documentation is complete and comprehensive. The consolidated project will be submitted as a zip file with a ReadMe.doc file listing assumptions (if any) made at your end and SQL scripts files (.sql) containing database and table definitions. If local servers such as XAMPP are used to test the project, include screenshots of working pages. Include port settings and other details in ReadMe file. In addition, you must submit a video clip showing the actual working of the Web application. Over and above the given specifications, you can apply your creativity and logic to improve the application