TEST Engineering & Management

May-June 2020
ISSN: 0193-4120 Page No. 4642-4646

Automated Life Management Application

Vinay Kumar[1], Priya Kothari[2], Rashmi U[3], Rachana Patil[4]
[1]Assistant Professor, [2,3,4]Student, C and IT department, Reva University, Bangalore, India
[1]vinaymkv@gmail.com, [2]priyakotharichiku@gmail.com, [3]rashmiuday16@gmail.com,
[4]patilrachana15@gmail.com

Article Info
Volume 83
Page Number: 4642-4646
Publication Issue:
May-June 2020

Article History
Article Received:19 November 2019
Revised: 27 January 2020
Accepted: 24 February 2020
Publication: 12 May 2020

Abstract

In the busy life schedule, people find difficulty in managing their personal and professional details which are needed in certain situations and retrieving them instantly will be a time consuming task without which fatal consequences would be faced. It is very helpful to have some means to access to one's health status and other aspects like blood group. One finds it difficult to manage the income sources and expenditure in their day to day life. In this regard, we have planned to design a web application that can store all the necessary information of an individual on one platform that can help to store and retrieve such information as & when required with safety and privacy measures implemented. This web application can be helpful in situations like when a person goes to bank to open a bank account, he would require necessary details such as Aadhar-card, PAN card etc. This web application can be accessed in any system and all the necessary documents can be shown right away instead of carrying them.

Keywords: *FTP- File Transfer Protocol, HTTP- Hyper-Text Transfer Protocol, OATH- The Initiative for Open Authentication*

1. Introduction

A healthy lifestyle is one of balance, meaning maintaining necessary self-care while also accomplishing daily tasks which need to get done – work, taking care of children, pets, and all other items which may popup throughout the typical day.In the modern lifestyle of today's generation people are engaged with a lot of activities which is taking a toll on their health. Due to this rush in the event of activities people are finding difficult to manage their lifestyle. It is necessary to manage it in a proper way for an organized& healthier lifestyle.

In this context, we propose an idea to have all the necessary information of an individual in an organized way on an application and can be accessed immediately. All the personal and professional details under one basic application that includes user data such as bank account details, health reports, important passwords and documents, sources of income and expenditure, helps the individual in an organized and healthier lifestyle.

2. Different Components in the Application

Data to be stored on the application is divided into different

Segments:

- **Income and Expenditure Tracking**– One can manage the daily expenses according to the income sources and can set expenditure limits. Hence, with the help of this application expenses are controlled and can encourage the person in future savings.
- **Health**– The user can store all the important documents related to health. If a person is suffering from any disease, he can store all the past and present reports in this section and can refer when necessary without any physical activity. A feature is implemented which will remind the user to take regular medications (if any) by sending an e-mail or text message.
- **Bank Details**– In this section, the user can store all the necessity bank details of different accounts. It includes the account holder's name, account number, IFSC code, Branch name and the related card details.
- **Passwords–** We can also store all the required e-mail id - passwords, passwords of social media websites with extra security.
- **Documents–** This part of the application can store scanned important documents for reference such as Birth certificate, SSLC / II PUC marks card, PAN card, Aadhar card etc.

- **Calendar-** To store the important dates and which can store necessary tasks to do or a particular day as remainders.

When it comes to storing personal information, security becomes at most priority. The data is secured by sending an OTP to the user's registered phone number.

3. Stages of Developing a Web Application

Following are the seven main steps:
1)Information Gathering: This is the stage of discovering and researching.It also determines how the following steps would look like.We need to get clear understanding of purpose of our application and the objectives we want to acquire, and therefore target the audience you wish to attract to your site.

2) Planning**:** At this stage, the developer creates the data and checks how the complete site will appear on the screen. Step by step procedure of creating the application is decided in this stage. Proper planning is needed to develop a webpage. We should also specify the relations between the main areas of our website.

3) Design: In this phase, the visual contents, such as images, videos or infographics is created. Keeping the purpose of the application in mind, the designer develops the website for the target audience. The layout of the website is the result of a designer's work. Layouts contain color theme, framework and images that provides a general understanding of the future product.

4) Content Writing and Assembly: Here, it is necessary to place in writing by understanding the audience and knowing their points and what they want. Here you'd prefer to connect to the audience of our website and add CTA(Call-to-Action). This stage also involves the creation of catchy headlines, compiling the existing text, text-editing etc.

5) Coding: This step is mainly to start developing the web application by referring to the results of the previous steps.It is designed using the specified programming languages for successful working of the application.

6) Testing, Review and Launch: Testing is one of the important part of creating a web application. Every single link must be tested to make sure that there are necessary connection among them and no broken ones. We should check all the scripts and forms, to check if the required output is achieved. Usage of code validators is done to check if our code follows the present web standards. Validating the code is mandatory.

7) Launch: After re-checking the application, we need to launch our application. It's time to upload it to a server. An FTP software is utilized for that purpose. After we deploy the files, we should again run again for final test to be sure that all our files have been installed correctly.

8) Maintenance: It's not just enough to "deliver" a website to a user but also make sure that everything works fine, and the target customers are satisfied and always be prepared to enhance the application anytime. To detect possible problems the end-users face, a Feedback system is added to the site for the benefit of the users.

4. Project Modules

Project is done as 3 modules:-
1. First module is about designing the front-end application using Hypertext Markup Language(HTML/HTML5) , the look and feel of the front end is designed using the cascading style sheets(CSS/CSS3).
2. Second module is about storing the data in an organized way about the individual using Oracle SQL developer.
3. Third module deals with modifying, retrieving and processing the data depending upon the requirement and is achieved through JAVA programming language.

Implementation of the project is through a web application using HTML for front end application development and to reduce the load on the server we are validating the data on the client side itself using a scripting language (JavaScript).

Data is stored on the light weight database, ORACLE Database and JAVA programming language for coding.

HTML - HTML is utilized to create electronic documents such as web page that are displayed on the World Wide Web (www). It is used by the browser to manipulate images,text, layout and other contents to be displayed in required format. We can say features like web-page formatting, images, video and audio elements designing, page layout techniques and graphics are included in HTML.

CSS–Cascading style sheets are to give page layout&well defined.
Visual feel&look to web pages and attract the users by making it more interesting and good looking. It is a language used to describethepresentation of a web page by formatting it's layout.

JavaScript- JS is a client-side programming language which helps a web developer to create a Web Application and make an interactive and dynamic web pages by implementing customized client-side scripts.

It can also perform an operation like when a user clicks on an image, or on a text, if it is described by the developer while creating the web application.

Java- We can create dynamic websites by using Java. It provides support for web applications through JSP's and servlets. With the help of static HTML pages, we can create a website.

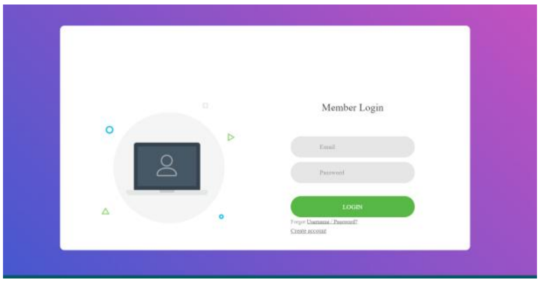
Java servlets-The servlet Technology makes use of Java language to create web applications. As it utilizes Java, the application made is robust, secured and scalable. They receive and respond to the requests from Web-clients, across HTTP.

Oracle Database- is a relational database management system used to store data for websites. It runs as a server and can allow multiple users to create and manage numerous databases. It is used for both small and large application. It is reliable, fast and flexible and easy to use. It supports standard SQL(Structured Query language).

5. Implementation Method

The front end development of the application is done using HTML, CSS, and JavaScript along with the JavaScript file if the special functionality is to be performed.

i) Login page: If the user has already created an account on the application then he can login using e-mail id and password or else needs to create an account page.



Member Login

Email

Password

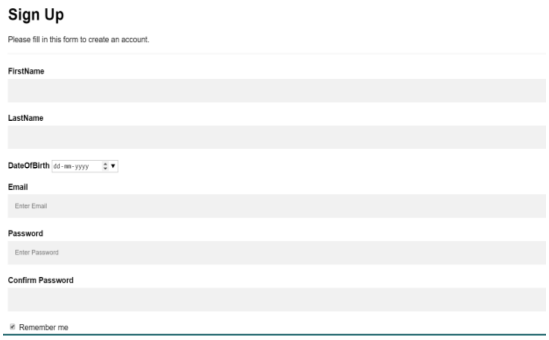
LOGIN

Forgot Username / Password?

Create account

Figure 1: Login Page

ii) Sign up page: If the user doesn't have an account then the user can sign up for the application on this page by giving the necessary details. After successful sign up, the control is redirected to login page.



Sign Up

Please fill in this form to create an account.

FirstName

LastName

DateOfBirth dd-mm-yyyy

Email

Enter Email

Password

Enter Password

Confirm Password

☑ Remember me

Figure 2: Sign-Up Page

iii) All the different components of the project are displayed, the user can select the required option and the page will be directed to the user's interest.

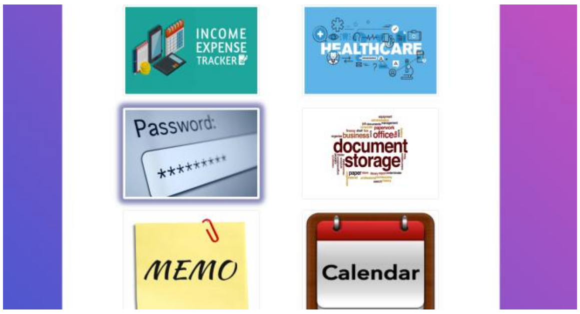
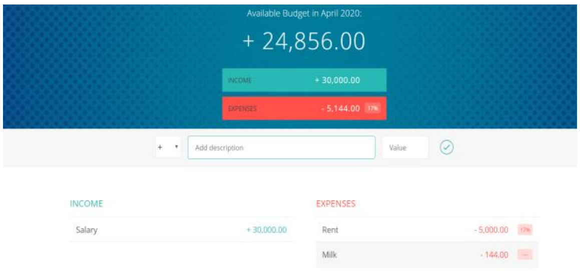


Figure 3: User's interest selection page

e.g1). If the user clicks on **income expense tracker** icon, the following page will be displayed.



Available Budget in April 2020:

+ 24,856.00

INCOME + 30,000.00

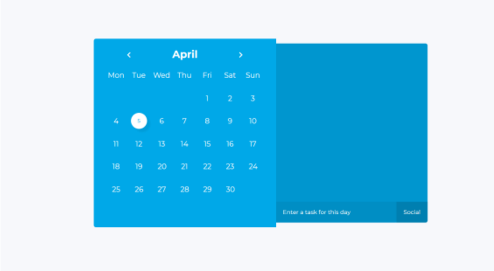
EXPENSES - 5,144.00 17%

+ Add description Value

INCOME		EXPENSES		
Salary	+ 30,000.00	Rent	- 5,000.00	17%
		Milk	- 144.00	---

Figure 4: income expense tracker

e.g2). If the user clicks on **Calendar** icon, the following page will be displayed,



‹ April ›

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Enter a task for this day

Social

Figure 5: Calendar

After the front-end development of the application and linking all the pages, the back-end is developed. We need to establish the connection initially so that the data can be stored and operated in the database. Back-end is also known as the server-side part of the application that allows it to operate but cannot be accessed by the user. Here we are using the oracle SQL developer that is an Integrated Development Environment(IDE) that provides end-to-end development of the application.

Client Server Architecture

Client Server Architecture is called as a computing-model which shows the relation between client and the server. Here, the server hosts, manages and delivers most of the services and resources to be consumed by the client. This type of architecture can have one or more client computers over an internet connection that is connected to a single central server.

Client–server **architecture** divides the application into 2 parts i.e., 'client-side' and 'server-side'. Such an application is implemented which establishes connection between the client and the server on a computer network. Front-end is known as the client-side, it is what the users see and Back-end is known as the server side, it is actually how everything works. Client-server can be called as a system that performs both the functions of server and client so as to promote the sharing/transfer of information from client to server or vice-a-versa. It allows multiple clients to have access to the same database at the same time, and the database will store much information.

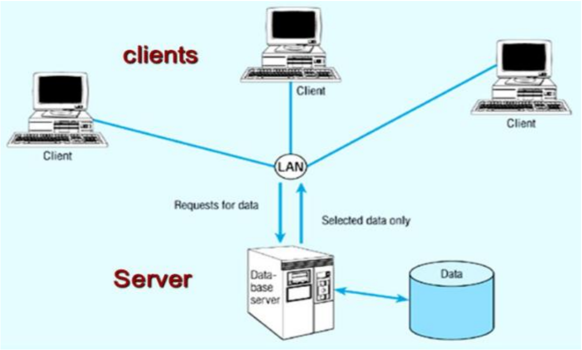


Figure 6: Client server architecture

We can observe the position of Servlets in a Web Application shown in the following figure.

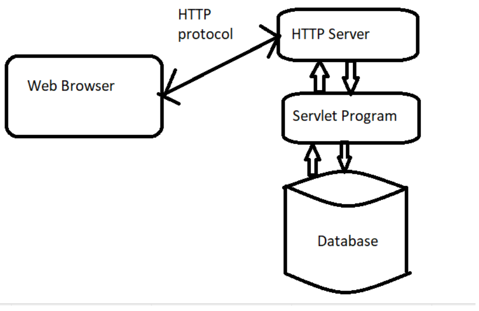


Figure 7: Servlets architecture

The servlets initially reads the explicit data that is sent by the Client (web-browsers) that may include a HTTP client program. It then reads the implicit HTTP request data that includes media types, cookies and the schemes the browser understands which is sent by the client. After which it processes the data that may require operating the database or invoking a web service and gives the result. Later the explicit data is sent to the client in the required format. It sends the implicit HTTP response to the client by telling the browser the type of format say document is being returned, catching parameters and other such tasks.

6. Security Measures

The rise in the cyber security threats is a challenge for every web applications. It has become very necessary to increase the security standards of our web application. We have to make sure that our user's accounts and their personal details are secured.

The basic method of authentication over the internet is ID and Password. But it is also a dangerous method to provide protection against attacks which can be replay attack or eavesdropping. We can overcome this problem with the help of HOTP algorithm that is based on 1-way hashing function SHA-1 which is the most popular OTP generation technique.

The application is added with an extra layer of security other than the mail-id and password in the login page by enabling **2-factor authentication** by implementing **TOTP** (the Time-based One-Time Password algorithm) authentication method. After enabling 2-factor authentication, the user after validating the e-mail id and password has to go through one more step of OTP verification to log in successfully.

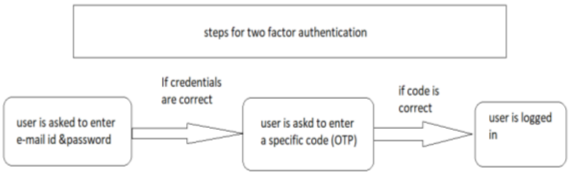

steps for two factor authentication
If credentials are correct
if code is correct
user is asked to enter e-mail id &password
user is askd to enter a specific code (OTP)
user is logged in

Figure 8: factor authentication

Now, an unauthorized third party cannot access the user's account without the access of both password and OTP.

One Time Password (OTP) algorithm in Cryptography

OTP: OTP is a unique numeric code that is randomly generated during every authentication event. A fresh set of digits is generated randomly as OTP each timean authentication is attempted. The generated OTP is unique during every authentication event.

Advantages:

i. An additional layer of security is added along with the username and password.

ii. The quality of being **unpredictable**.

The 2 main methods for OTP delivery are:

1.) SMS Based: (OTP via a text message)

2.) Application Based: (scanning QR code on the screen)

We are going to implement message-based OTP generation method which is defined by OATH i.e., the TOTP, which is a Time Synchronized OTP.

HMAC-based One-time Password algorithm (HOTP) is an OTP algorithm which is based on hash-based message authentication code(HMAC). The HOTP algorithm is an authentication method by the symmetric generation of human-legible values which can be used for only one authentication attempt.

Purpose:_ It can be used to verify simultaneously both the integrity of data and the authenticity of a message.

HMAC uses 2 passes of hash computation: The **secret key is first used to derive the two keys.** The 2 keys are inner and outer.

→**First pass -** Here, the algorithm derives an internal hash that's originated from the message along with the inner key.

→**Second pass** - This pass produces the final HMAC code that's originated from the inner hash result along with the outer key.

A message is broken using an iterative hash function into fixed size blocks and iterates over them with the assistance of a "**compression function".**

$$\mathrm{HMAC}(K,m)=H((K' \oplus \mathrm{opad}) \,||\, K(K' \oplus \mathrm{ipad}) \,||\, m))$$

$$\begin{aligned} &\mathbf{K' = H(K)} \quad \mathbf{if\ K > block\ size} \\ &\mathbf{K' = H(K)} \quad \mathbf{otherwise} \end{aligned}$$

Snapshot-1: Hash function equation

Where,

H is a cryptographic hash function

m is the message to be authenticated

K is the secret key

K' is a block-sized key derived from the secret key, (*K*; either by padding to the right with 0's up to the block size, or by hashing down to less than the block size first and then padding to the right with zeros)

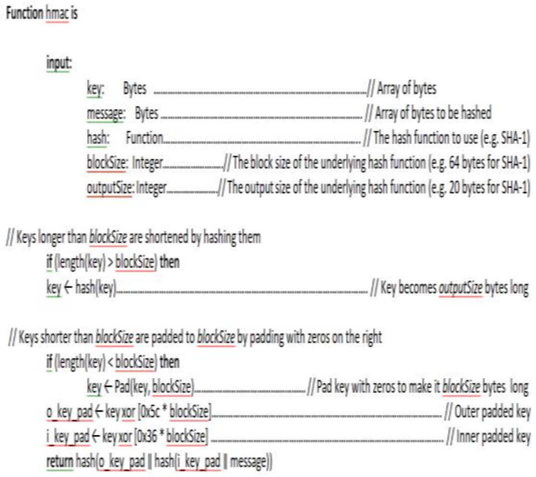
|| denotes concatenation

⊕ denotes bitwise Exclusive OR (XOR)

opad is the block-sized outer padding, consisting of repeated bytes valued 0x5c

ipad is the block-sized inner padding, consisting of repeated bytes valued 0x36

Pseudocode to implement hash function using Sha-1 hash function



```
Function hmac is

    input:
        key:     Bytes ..........................................// Array of bytes
        message:  Bytes .........................................// Array of bytes to be hashed
        hash:    Function........................................// The hash function to use (e.g. SHA-1)
        blockSize: Integer...............// The block size of the underlying hash function (e.g. 64 bytes for SHA-1)
        outputSize: Integer..............// The output size of the underlying hash function (e.g. 20 bytes for SHA-1)

// Keys longer than blockSize are shortened by hashing them
    if (length(key) > blockSize) then
    key ← hash(key)...............................................// Key becomes outputSize bytes long

// Keys shorter than blockSize are padded to blockSize by padding with zeros on the right
    if (length(key) < blockSize) then
        key ← Pad(key, blockSize).....................// Pad key with zeros to make it blockSize bytes  long
    o_key_pad ← key xor [0x5c * blockSize].................................// Outer padded key
    i_key_pad ← key xor [0x36 * blockSize] ................................// Inner padded key
    return hash(o_key_pad ‖ hash(i_key_pad ‖ message))
```

Snapshot-2. Illustration of hash function

7. Conclusion

A user can store all the personal information in one place and is easily accessible. It can be accessed at anytime and anywhere. It can save a lot of time in searching certain personal information in different methods. It comes in handy for a lot of people who have a busy and hectic schedule in their day to day life. All the important personal data can be accessed in this application with at most security provided and need not worry about the loss of data. Thus this web application helps a lot of people who find difficulty in managing their lifestyle.

Expected new implementation:

- To provide higher security by sensing the fingerprint and then sending the OTP.
- Improvement in the features of the application.
- To develop it into a mobile application.

8. Acknowledgment

We are extremely thankful to our guide **Prof. Vinay Kumar** for the guidance and help during the development of our project. He always gave us useful suggestions and also helped us when the work was not moving ahead at times.

References

[1] Haroon Shakirat Oluwatosin, "Client-Server Model" -2014,IOSR Journal of Computer Engineering.

[2] Somnath Mazumdar, Daniel Seybold, Kyriakos Kritikos &Yiannis Verginadis , " A survey on data-storage" -2019.

[3] Ch Rajesh, K S V Krishna Srikanth, "Research on HTML5 in Web Development" -2014.

[4] Young Sil Lee, HyoTaek Lim, HoonJae Lee, " A Study on Efficient OTP Generation using Stream Cipher with Random Digit" -2010