



THE JOURNAL OF COMPUTER SCIENCE AND ITS APPLICATIONS

Vol. 27, No. 1 June 2020

MODEL IMPLEMENTATION OF TEXT AND VIDEO CHATS WITH PYTHON IDE SOFTWARE

R. O. Bello¹, M. A. Ogunrinde² and O. B. Bello³

¹*Department of Computer Science, University of Ibadan, Ibadan, Nigeria*

²*Department of Mathematical and Computer Sciences, Fountain University Osogbo, Nigeria*

³*Department of Agronomy, Federal University Gashua, Yobe State, Nigeria*

¹*belloridwan50@gmail.com* ²*bogunrinde@gmail.com*, ³*obbello2002@yahoo.com*

ABSTRACT

Chats are bound to be an efficient way of disseminating synchronous and asynchronous information through textual messages or video communication among various people over a network. An Integrated Development Environment (IDE), is a text editor that allows developers and engineers to enhance the different features of writing executable software. Using IDEs, the programmer's productivity is enhanced by consolidating common activities involved in writing computer programs into a single application. These activities include source code editing, auto-completing, syntax highlighting, building executable, debugging, and automated testing. Python is a programming language used to develop mostly sophisticated Machine Learning and web applications. An IDE makes coding in Python much easier and efficient. It consists of features like indentation and code folding which are very important aspects in executing Python programs successfully. Also, unresolved imports or undefined methods are indicated with warnings and error signs to give programmers more information on impending problems. A web-based Python Integrated Development Environment (IDE) was developed by applying modern communication techniques of text and video chats. This software delivers facilities to programmers for efficient development of programs. It also consists of auto-completion, code fixing, code editing, management and automated testing. A cloud-based IDE was deployed because it enables programmers to easily access their projects anywhere and at any time. Also, hardware limitation issues are eliminated with no setups and configurations required, as all the information is to be stored in the cloud. Hypertext Markup Language, Javascript, and Bootstrap Cascading Style Sheet were used to design the front-end interface, while Asynchronous Javascript, and Extensible Markup Language, Hypertext Pre-processor and My Structure Query Language were employed for the back-end design. Various users can be able to write, edit, debug, run, and store the codes on the server. The user can also use the IDE on any internet-enabled devices such as personal computers, mobile devices, and tablets.

Keywords: Synchronous, integrated development environment, interface, online, communication.

1.0 INTRODUCTION

The importance and popularity of chat applications in a software development

environment have been increasing every day [1]. This is as a result of a lucrative solution which enables seamless interaction among

programmers concurrently. The real-time interactions will allow other members of the group to communicate and solve problems together anywhere at any time. Chat applications are mostly integrated with the development environment [2]. An Integrated Development (IDE) consists of tools existing in isolated or separated systems, including a code editor, a debugger, and a compiler with all these disjointed components integrated to make a complete "integrated" system. Nevertheless, all IDEs have yielded tremendous growth over the past ten years and have significant advancement in all functionalities which have only fortified their wide acceptance by the target audience.

Developers hold a conversation in online chat rooms in real-time with the use of the Internet and their personal computers to send text messages and show in the receivers' screens in real-time. This chatting platform offers a leeway in understanding the underlying problem very well. Due to its synchronous nature, the textual chat is deployed to augment face-to-face interaction with advantages including repair moves and negotiation for meaning [3]. For example, it was discovered that members in the synchronous chat utilized interactional alterations analogous to face-to-face interactions. The research revealed that most people make use of textual chat to initiate communication and organize different activities [4]. Texting offers enormous importance to explore various tasks and chat in real-time [4]. Studies also showed that textual chat does not cover all aspects of face-to-face communication [5]. Text chatting can accommodate many participants that can forward messages concurrently concerning similar or discarded topics. Therefore, this produces a sequence of dialogues apart from the face-to-face interaction in which members of the group usually talk on a single thread at a time. Given this, the textual chat platform may require the need necessary for repair moves because of interruptions in communication

due to topic incoherence [5].

Video chat or conferencing can be referred to as a real-time connection among people in disparate sites for interaction and communication often consists of audio and video [6]. This offers the transmission of text and static images between different sites. It also provides the transmission of high-quality audio and full-motion video images among many sites [7]. Video conferencing technology was invented about 40-50 years ago. The advancement of video conferencing relied on the accessibility of a dependable digital communications network. Video conferencing has made significant improvements in communication between remote locations, both within the industry and between the learners and tutors. This also speeds up the development of the project. Users from different locations can access, send, and make submissions just as e-mail as transformed the sharing of information. Also, video chat allows for video sessions to be recorded for later reference, involving promotional or instructional activities. A live video conference may be developed by delivering video conferencing software resources on a machine such as a strong Internet connection, microphone, and a webcam [8]. These communication resources offer wide-ranging services of information sharing, asynchronous and synchronous interactions among team members. Video chatting promotes awareness of activities done by other team members in real-time, artifacts changes, and prevents coding errors as well as mistakes [9]. In contrast to the audio-only calls, a video conferencing application allows non-vocal communication described to be an efficient approach to the collaborative environment [10].

Real-time Communication (RTC) is an amalgamation of communication and collaboration systems, which is a combination of communication technologies, like Voice-over-IP (VoIP) telephony, instant messaging, and various collaborative applications [11]. RTC systems usually

enable two-person communication and support the multi-person conferences. By providing and integrating a range of synchronous communication media in one integrated environment. RTC systems allow users to collaborate in real-time by editing a document, voice call, multi-person, and video conference. RTC technology could assist in solving synchronization challenges, especially when working as a team. It offers flexibility and interacting remotely with other users. RTC system has been facilitated and improved on users' interaction and communication for effectiveness [12].

Python is an open-source and object-oriented programming language developed by Dutchman Guido van Rossum in the 1980s [13]. The language can be utilized for a wide range of applications such as scripting, developing, and testing. Given its simplicity, large technology companies which include Hewlett-Packard, Dropbox, Quora, Mozilla, Qualcomm, and Google have deployed Python in their applications.

Python was adopted for this research because it is one of the most popular dynamic programming languages being used currently. In June 2017, Python was the most visited tag on Stack Overflow and ranked the top 2 in high-income nations including the US and the UK (next to either Java or JavaScript) [14]. Stack Overflow reported that the volume of visitors to questions about Python on their site is increasing more rapidly than any other, giving Python a claim to being the fastest-growing major programming language in the world [15]. Further, Python's popularity in Data Science and Machine Learning is one of the main drivers of its fast growth and versatility. There are over 125,000 Third-Party Python Libraries worldwide. These Libraries enable Python to be useful; ranging from Django and Numpy for web development and text processing, respectively, to the cutting edge such as Scikit-learn and Pandas for Machine Learning and Data Science, respectively [16].

The objective of this study was to design and implement text and video chats with python Integrated Development Environment software with the view of creating a tutorial environment that would support remote collaboration between two users in real-time.

2.0 RELATED WORK

PythonAnywhere is a cloud-based online service and IDE where programs written in Python programming language can be compiled, maintained, and executed [17]. It allows teams at dispersed areas to engage in a collaborative environment in the implementation of programming methodology used for comprehensive software development systems. It offers access to a cloud-based Git Bash command-line interface with Python using an editor including highlighting of syntax. The software documents may be moved to and removed from the web service via the client's browser. Applications running on the web may be published on a Web Server Gateway interface-based application framework. This is illustrated by the modest approach in deploying web2py applications on the server in the official book on the web framework. It is used to study Numpy, Pandas, Matplotlib, and endorsed as an approach to presenting machine learning applications on the web.

The interactive Python [18] enables tutors to develop animated books along with the program code, data visualizations, videos, polls, form-based questions, comment boxes, code-based questions, and other valuable characteristics online with the contents. Online Python Tutor [19] is an online technology that assists learners to comprehend software programs by creating a visual image of the stack frames equivalent to each statement. The technology functions for programs developed in Python and other programming languages such as JavaScript, Java, Ruby, and TypeScript.

Remote Application Platform is an open-source software code editor developed using the Eclipse Technology Project. The approach aims at enabling designers to design Ajax-enabled rich online applications with the use of Eclipse development architecture, and Java application programming interface (API). The platform may also be denoted as similar to web application development on the Rich Client Platform (RCP) [20]. Python Fiddle, on the other hand, is a development platform deployed to write and run program codes, enables developers to fix bugs and scripts as well as running code snippets online. It includes the promotion of a large number of third-party libraries, provides interactive and explainable documentation, as well as providing an extensive pool of in-built keys [21].

Code Anywhere is a code editor executed on the web browser with the File Transfer Protocol (FTP) client and provides support for tools used in the web involving JavaScript, Cascading Style Sheet (CSS), Extensible Markup Language (XML), Hypertext Markup Language (HTML), and PHP. [22]. Kodingen is also an online IDE run on the web to provide a faster, easier approach to create, implement, and deploy applications on the web with the use of numerous frameworks and languages. This allows for a wide range of programming IDEs and tools [23]. Recently, the NodeJS framework has been a leeway to incorporate server-side programs on the web. This is a framework that allows users and developers to develop and deploy server-side codes on the client-side with JavaScript scripting language. Specific web development environments that use NodeJS as a tool include AkShell [24].

An online compiler or interpreter can be designed as a cloud computing concept called Software as a Service (SAAS). The major aim is to make writing, debugging, compiling, and executing codes online.

Cloud-based compilers and interpreters always help to minimize the memory space and to make it readily available. The cost of development and time can be decreased greatly when cloud computing technologies are used for the programming environment. It is a highly appropriate tool for compiling program codes, detecting and eliminating program errors, and debugging codes [25]. IDE 2.0 is a collaborative tool for software engineering which provides services that inspired a collaborative environment on Web 2.0 application software. This suite is considered as the instantaneous editor deployed for teams in collaboration with running source code documents [26].

This research work is expected to estimate the programming techniques in Python with no installation of a compiler or dependencies required to run the program efficiently. The student can launch their favorite browser and enter the Uniform Resource Locator (URL) of the website. They may also be asked to enter their credentials for login authentication, and if the status is successful, they will be redirected to a page where the student can write, compile, and run Python codes. Nevertheless, if the student does not have the necessary permission or dependencies to install the software on their machines (such as working on a computer system in a computer Laboratory as a non-administrative user), which will not allow the application to run efficiently since the necessary components in this type of software development model are incomplete. This setback creates an important blocking issue that may not easily be overpowered by simply replacing them or changing the underlying software development technologies or tools. As only the online IDE via the design and development model cannot elucidate this problem, the editor was hosted on the cloud in this study. However, the overall application was further enriched with the integration of live video chat and textual chat for synchronous communication among the collaborators to facilitate an easy definition

of the underlying problem.

3.0 IMPLEMENTATION

The conceptual design of the system provides an overview of the research by applying System Architecture, *System Flowchart*, and *Use Case Diagram of the System*.

3.1 System Architecture

Users have to log in first to be able to access the workspace as depicted in Figure 1. User data will be checked first whether it has been stored previously in the database. If the data exist then new log data will be saved into Users logs, if not then new data will be created in user, file, and log Users. Moreover, the registered user will be readdressed to the workspace page, and the project file data will be fetched from the database. Other active members list will be loaded after the user

redirected into the workspace. This friend's list will be used to add new collaborators, and also new collaborators who have been invited will have the ability to communicate with other users via textual or video chat. Also, the user can send a chat to other users and adds collaborators to join the same file project. When the user types a message and press the enter key, the encoded data will be sent and then the chat data is saved into the chat Collection. Thereafter the data will be sent to the server and then data is returned to all members on the system. After getting the user id, data will be inserted into file Collection and will be forwarded into the server. Meanwhile, the System Architecture of the proposed system is depicted in Figure 1.

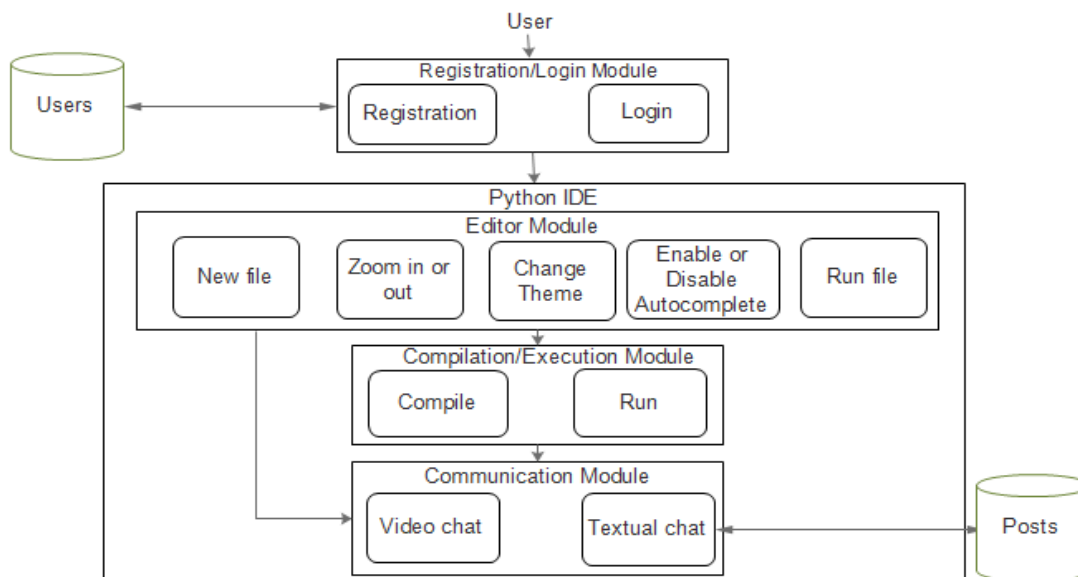


Figure 1: System Architecture

3.2 System Flowchart

It is a diagrammatic solution to the stated problem and the process operations are implied by the sequencing of operations as displayed in Figure 2.

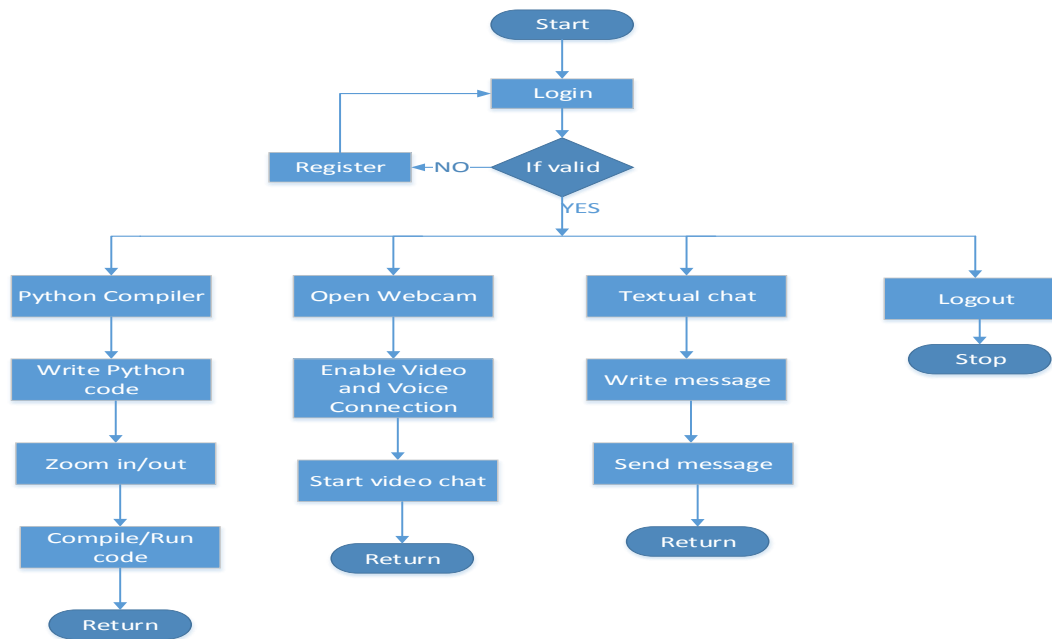


Figure 2: System flowchart

3.3 Use Case Diagram of the System

For the system to be accessible, the users will register with their personal information including their passport thereby getting the username and password used for registration. The registered user will be able to view all

the users currently in the system, write, compile and run Python codes and chat with them concerning the programming issues and finally pair with the member of the system for video conferencing as shown in Figure 3.

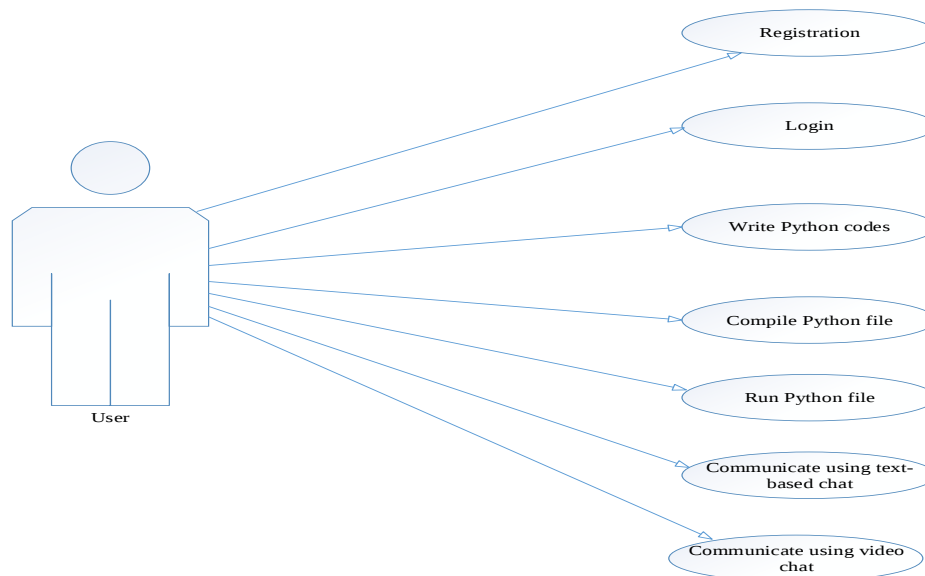


Figure 3: Use Case diagram of the system

4.0 RESULTS AND DISCUSSIONS

An online Python compiler with text and video chats was developed with the use of HTML, CSS, PHP, and AJAX as the programming tools. The extensive support for deploying PHP based applications on the internet encouraged the notion of building the application using PHP and AJAX technologies.

4.1 Home page

As the IDE is provided online, the users need to have an internet connection to access it. The users will have to type the appropriate Uniform Resource Locator (URL), while the home page will be displayed to the users as presented in Figure 4.

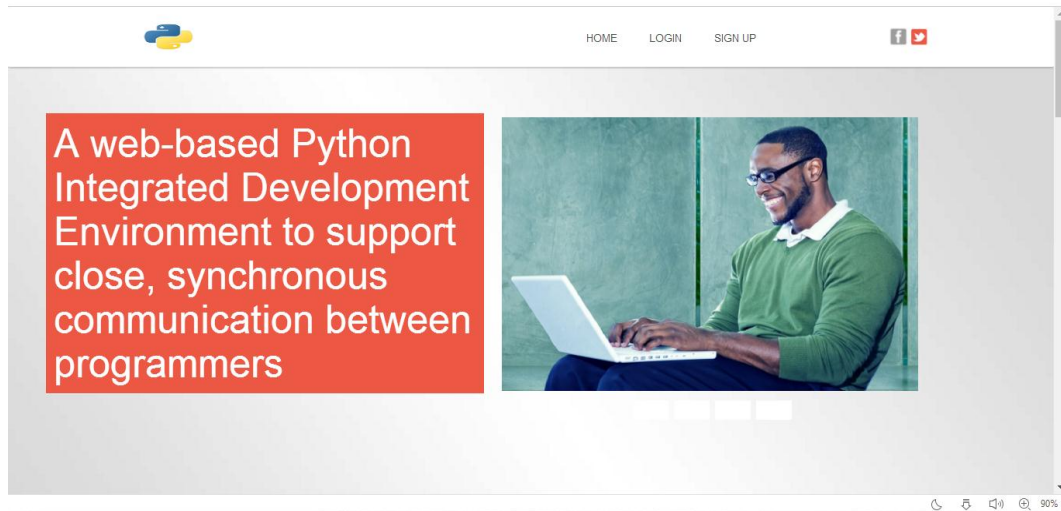


Figure 4: Home page

4.2 Registration Page

This registration page is a page on the web application which allows users to fill in their details including their

profile pictures. Details like email and password will be valuable when logging into the system as described in Figure 5.

The screenshot displays the 'SIGN UP' registration form. The form is centered on the page and contains several input fields: 'First Name', 'Last Name', 'E-mail address', 'Phone Number', 'Username', 'Password', and 'Confirm Password'. There is also a section for 'Upload Profile Picture' with a 'Choose File' button and a note 'No file chosen'. A red 'REGISTER' button is located at the bottom of the form. Below the form, there is a link that says 'Have an account already? [Login](#)'. The page has a consistent header with the application logo and navigation links, and a browser window footer at the bottom.

Figure 5: Registration page

4.3 Login Page

The login page enables users to fill in their usernames and passwords used during registration. This will give them access to

the Python editor page with text and video chats, as revealed in Figure 6.

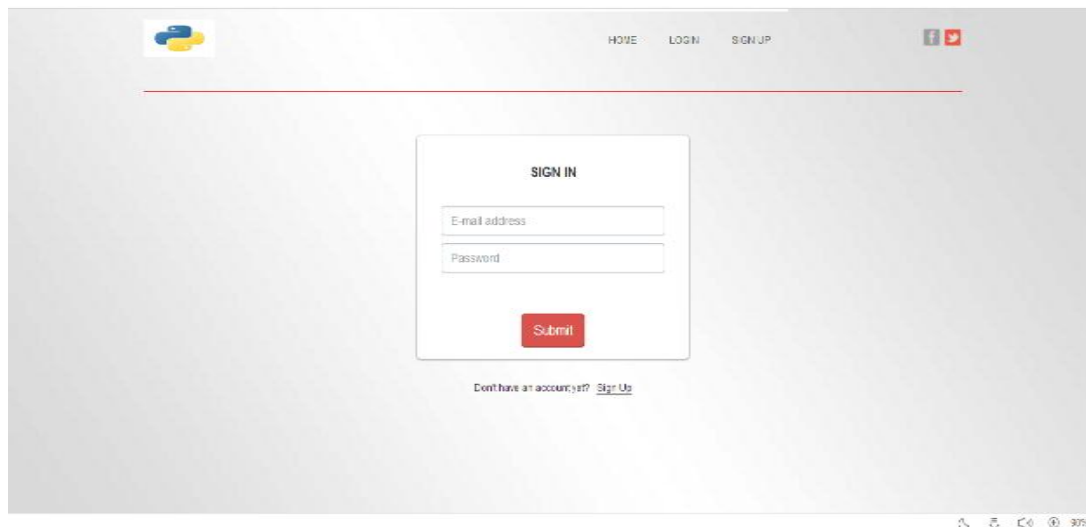


Figure 6: Login page

4.4 Python Editor with Text Chat Page

After a successful login, the user will be redirected to a page called the “programming panel”. This page consists of the Python editor, textual chat, list of users online, and the video chat. Users can click the button placed at the left corner of the workspace

page to access the folder and the files in the workspace as shown in Figure 7. Action is provided such as create, delete, zoom in, zoom out, autocomplete, theme change, compile, and run. The video communication can also take place on the same page as displayed in Figure 8.

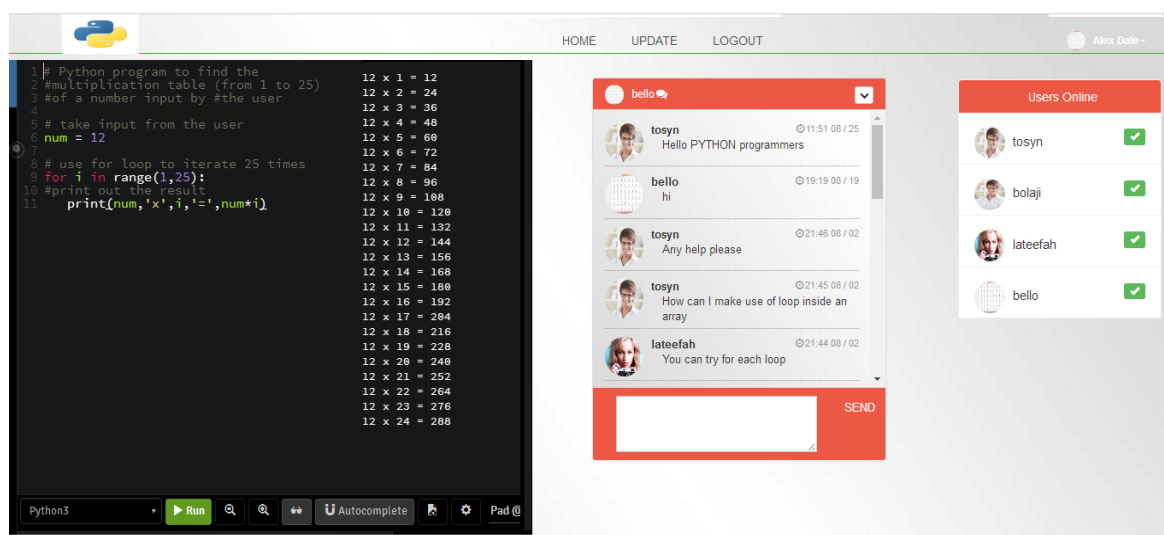


Figure 7: Python editor page with text chat

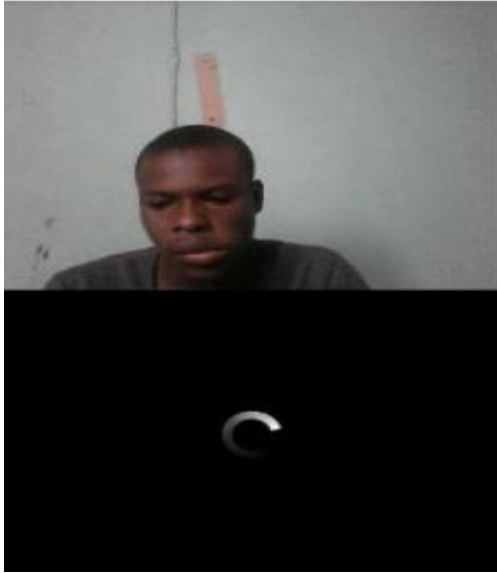


Figure 8: Video chat page

4.5 Edit Profile Page

In the edit profile page, users can view their details inserted during registration. They also

have the option to update it and also change the profile picture, as exhibited in Figure 9.

Figure 9: Update page

5.0 CONCLUSION

In this research work, we implemented a cloud-based Python IDE enhanced with state-of-the-art communication tools of

textual and video-based chats. Due to insufficient memory and Central Processing Unit cycle requirements by desktop-based IDE, we developed an online code editor that

takes s advantage of the facilities offered by the web such as none deployment, instantaneous access to services provided, and limited computational resources. These advantages allow the possibility of developing a chatting platform for users and programmers who encountered some computer-related challenges and are open to relate it with their counterparts. The combination of synchronous chats with lists of tasks in which programmers follow and can easily be accessible will give fellow project partners clear information on what they need to do which makes the work faster and organized. This study presents a promising phase towards achieving a design of a feasible, scalable, and robust software development environment for learning and teaching to program in Python programming language.

Appendix A: Methods for Sending and Receiving Messages

// Program to send message to database using AJAX

// declaration of javascript

```
<script type="text/javascript">
```

// function to get the message from the textbox

```
function text(){
document.getElementsByName('msg')[0].placeholder = "Enter your message here .....";
}
```

// end of the text function

```
</script>
```

```
<script type="text/javascript">
```

// function to submit the text into the database

```
function submit(){
// checking if the username and the message box are not empty
if(form1.username.value == " ||
form1.msg.value == ""){
    alert('ALL FIELDS ARE
REQUIRED');
    return;
}
```

```
    else{
var uname= form1.username.value; // storing
```

the name into a variable name

```
var msg= form1.msg.value; // storing the
message into a variable name
```

```
//varmsgtime= form1.msgtime.value;
```

```
var xmlhttp = new XMLHttpRequest(); //
create http request instance
```

```
    xmlhttp.onreadystatechange =
function(){
```

```
// use the class to create a function which
displays the content of insert.php when
successfully loaded
```

```
    if(xmlhttp.readyState==4&&xmlhttp.
status==200){
```

```
        document.getElementById('chatlogs')
.innerHTML = xmlhttp.responseText;
// the chatlogs from the db will be displayed
inside the chatlogsaction
    }
}
```

```
xmlhttp.open('GET','insert.php?uname='+una
me+'&msg='+msg,true);
```

```
xmlhttp.send(); //open and send the http
request
```

```
    }
document.getElementById('yourid').value =
"";
}
```

```
</script>
```

```
<script>
```

```
jQuery(document).ready(function($) {
$.ajaxSetup({cache:false});
```

// reloading the page after every 2 seconds

```
    setInterval(function(){ $('#chatlogs').l
oad('logs.php');},2000);
});
```

```
</script>
```

```
<script>
```

```
jQuery(document).ready(function($) {
$.ajaxSetup({cache:false});
```

// reloading the users online after every 1 second

```
    setInterval(function(){ $('#online').loa
d('online.php');},1000);
});
```

```
</script>
```

```
<?php
```

```
include 'connect.php';
```

```
session_start();
```

```
?>
```

```
<?php
// fetch the result directly from the database
$results=mysql_query("SELECT * FROM
chat ORDER BY id DESC");
//$passport = $_SESSION['passport'];
while($fetch=mysql_fetch_assoc($results)){
$names = $fetch['name'];
$info = $fetch['messages'];
$timer = $fetch['msgtime'];
$dates = $fetch['msgdate'];
$pasort = $fetch['passport'];
$datem=substr($dates,5);
$dater=substr_replace($datem, ' / ', 2, 1);
    echo'
<li class="left clearfix">
<span class="chat- img pull- left">
<imgsrc="userimages/"; echo $pasort. ' ?>"
alt="User Avatar" class="img- circle"
height="50px" width="50px" />
</span>
<div class="chat- body clearfix">
<div class="header">
<strong class="primary- font">'; echo
$names. '</strong>
<small class="pull- right text- muted">
<i class="fa- clock- o fa- fw"></i>'; echo
$timer. ' '. $dater. '
</small>
</div>
<div style="height:auto;"
class="col- xs- 12">
'; echo $info. '
</div>
</div>
</li>

';
}
?>
```

Appendix B: Video API

```
<script>
// getting the API key credentials
varapiKey = '45828062';
// storing the session inside a variable
varsessionId =
'2_MX40NTgyODA2Mn5- MTQ5ODk5NT
E0MjYzNH55OHJRN2RKWnpDcDsVdPV
EJHYWRybTRzWFV- UH4';
```

```
//storing the token ID inside a variable
var token =
'T1==cGFydG5lcl9pZD00NTgyODA2MiZz
aWc9NWUxZjAxNmRjYWVhZjNjMmRm
NzY2YTEzMDE1MjRkMzVjMzMzMzQx
MzpzZXNzaW9uX2lkPTJfTVg0ME5UZ3lP
REEdTW41LU1UUTVPRGs1TIRFME1qW
XpOSDU1T0hKUk4yUktXbnBEY1ZkUFZ
FSkhZV1J5YlRSeldGV1VSDQmY3JlYXR
lX3RpbWU9MTQ5ODk5NTIyOCZub25jZT
0wLjY5ODE2NjI5MDgzMzEzNTYmcm9sZ
T1wdWJsaXNoZXImZXhwaXJlX3RpbWU
9MTQ5OTA4MTYyOA==';
// connect to session
var session = OT.initSession(apiKey,
sessionId)
session.connect(token, function(error) {
// create publisher
var publisher = OT.initPublisher();
session.publish(publisher);
})
// create subscriber
session.on('streamCreated', function(event) {
session.subscribe(event.stream);
}); </script>
```

REFERENCES

1. TirusMuya, M (2013). Instant Messaging an Effective way of Communication in the Workplace. *arXiv preprint arXiv:1310.8489*.
2. Hairudin, A., Azurah, A., DewiNasien Y. and Cheah, T. (2016). P2P Audio and Video Calling Application using WebRTC. *ARPJ Journal of Engineering and Applied Sciences*, 11(3), 1766-1770.
3. Emmadi, S. and Sirisha, P. (2019). Android-based instant messaging application using firebase. *International Journal of Recent Technology and Engineering*. 7(12). 352-355.
4. Kelly-Ann, A., Tracii R., DeLeon, D., McInerney, D. and Waters L. Social Media Use and Social Connectedness in Adolescents: The Positives and the Potential Pitfalls. *Australian Journal of*

- Educational and Developmental Psychology*. 31, 18-31. 10.1017/edp.2014.
5. J. W. Hameed, J. and AbuSeileek. A. (2015). "Using Text and Voice Chat Modes to Enhance Students' Performance in Discourse Functions." *International Journal of Learning Technology*, 10(2), 114-136.
 6. Daim, T., Ha, A., Reutiman, S., Hughes, B., Pathak, U., Bynum, W. and Bhatla, A. (2012). Exploring the Communication Breakdown in Global Virtual Teams. *International Journal of Project Management*, 30(2), 199-212.
 7. Denstadli, J., Julsrud, T. and Hjorthol, R. (2012). Videoconferencing as a Mode of Communication: A Comparative Study of the Use of Videoconferencing and Face-to-Face Meetings. *Journal of Business and Technical Communication – J BUS TECH COMMUN*. 26. 65-91. 10.1177/1050651911421125.
 8. Comer, J, et al. (2015). Rationale and Considerations for the Internet-Based Delivery of Parent-Child Interaction Therapy. *Cognitive and Behavioral Practice*, 22(3), 302-316.
 9. Thapa, M. (2015). Individual Adjustment of Audio and Video Properties in Network Conferencing. *U.S. Patent No. 9, 060,094*.
 10. Hsiao, E. (2012). Synchronous and Asynchronous Communication in an Online Environment: Faculty Experiences and Perceptions. *Quarterly Review of Distance Education*, 13(1), 15-18.
 11. Edim, A. and Bakwa, D. (2017). A Peer-To-Peer Architecture for Real-Time Communication Using WebRTC, 4(2), 2394-4404.
 12. Zinah, N., Sarah A. and Zena, H. (2019). Peer to Peer Multimedia Real-Time Communication System based on WebRTC Technology. *International Journal for the History of Engineering & Technology*, 2(9), 125-130.
 13. Diamond, S. and Stephen, B. (2016). CVXPY: A Python-embedded Modeling Language for Convex Optimization. *The Journal of Machine Learning Research*, 17(1), 2909-2913.
 14. David, R. (2018). The Incredible Growth of Python, Retrieved from <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>. [Accessed August 2018].
 15. Prince, P. (2018). Why Python is the Most Popular Language Used for Machine Learning. Retrieved from <https://medium.com/@UdacityINDIA/why-use-python-for-machine-learning-e4b0b4457a77>. [Accessed August 2018].
 16. PythonAnywhere. (2018). Host, run, and code Python in the cloud! Retrieved from <https://www.pythonanywhere.com/> [Accessed August 2018].
 17. Percival, H. (2014). Test-Driven Development with Python: Obey the Testing Goat: Using Django, Selenium, and JavaScript. *O'Reilly Media, Inc., 2014*.
 18. Runestone Interactive: Interactive python, (2018). Retrieved from <http://interactivepython.org/> [Accessed August 2018].
 19. Online Python Tutor (2018). Retrieved from <http://www.pythontutor.com/> [Accessed August 2018].
 20. Jayaraju, P. and Prakash, V. (2015). Survey on Web Based Interface. *International Journal of Advance Research in Computer Science and Management Studies*, 3(9), 243-247.
 21. Python Fiddle, (2018). Retrieved from www.pythonfiddle.com. [Accessed August 2018].
 22. CodeAnywhere, (2018). Retrieved from <https://codeanywhere.net>. [Accessed August 2018].

23. Kodingen, 2018. Retrieved from kodingen.com, [Accessed August 2018].
24. Web IDE, 2018. Retrieved from akshell.com. [Accessed August 2018].
25. Surya, C., Durga K. and Sudha, R. (2015). Online C, C++ and Java Compilers Using Cloud Computing. *International Journal of Computer Science and Mobile Computing*, 4(8), 348-355.
26. Zakaria, I., Noura, A., Anouar, A. and Kamal, E. (2014). Building a Web-based IDE from Web 2.0 Perspective. *International Journal of Computer Applications*, 96(22), 77-89.