A Project Report on

"AN APPLICATION FOR MSBTE AFFILIATED COLLEGES"

Submitted by

MOHIT HEMANTH SHETTY
SHAIKH MOHAMMED TALHA SALIM
MEHTA PARTH RAHUL
DAVE MAITRY NIMESH

Project Guide

Mrs. Vaishali Rane

DEPARTMENT OF COMPUTER ENGINEERING

ZAGDU SINGH CHARITABLE TRUST (REGD.)

THAKUR POLYTECHNIC

(An ISO 9001:2015 Certified Institute**)**

Thakur Complex, West to W.E. Highway, Kandivli (E), Mumbai – 400101.

2020-2021

Zagdu Singh Charitable Trust's (Regd.)
THAKUR POLYTECHNIC
(Approved by AICTE, Recognised by Govt. of Maharashtra & Affiliated to MSBTE)
(Accredited by : National Board of Accreditation, New Delhi*, ISO 9001:2015 Certified Institute)
Thakur Complex, Kandivali (East), Mumbai - 400101.
Tel.: 2854 2481 / 2854 3540 / 2854 7707 / 67756300 / 301 / 302 / 303 • Fax : 2854 1993
E-mail : tpoly@thakureducation.org • Website : www.tpolymumbai.in
Estd. 1998
*1st Time Accrediated Programmes : Mechanical Engineering | Electronics & Tele-Communication Engineering | Electronics Engineering (w.e.f-15-03-2012 for 3 Years)
*2nd Time Accrediated Programmes : Mechanical Engineering | Electronics & Tele-Communication Engineering | Electronics Engineering | Computer Engineering | Information Technology (for 2 Years upto 30-06-2019)

PROJECT APPROVAL SHEET

Academic Year 2020-2021

This Project work entitled

"AN APPLICATION FOR MSBTE AFFILIATED COLLEGES"

By

MOHIT HEMANTH SHETTY
SHAIKH MOHAMMED TALHA SALIM
MEHTA PARTH RAHUL
DAVE MAITRY NIMESH

Is Approved for the award of the

DIPLOMA IN

COMPUTER ENGINEERING

(Mrs. Vaishali Rane)

PROJECT GUIDE/MENTOR

---------- ----------

EXTERNAL EXAMINER **INTERNAL EXAMINER**

Zagdu Singh Charitable Trust's (Regd.)
THAKUR POLYTECHNIC
(Approved by AICTE, Recognised by Govt. of Maharashtra & Affiliated to MSBTE)
(Accredited by : National Board of Accreditation, New Delhi*, ISO 9001:2015 Certified Institute)
Thakur Complex, Kandivali (East), Mumbai - 400101.
Tel.: 2854 2481 / 2854 3540 / 2854 7707 / 67756300 / 301 / 302 / 303 • Fax : 2854 1993
E-mail : tpoly@thakureducation.org • Website : www.tpolymumbai.in
Estd. 1998
*1st Time Accrediated Programmes : Mechanical Engineering | Electronics & Tele-Communication Engineering | Electronics Engineering (w.e.f-15-03-2012 for 3 Years)
*2nd Time Accrediated Programmes : Mechanical Engineering | Electronics & Tele-Communication Engineering | Electronics Engineering | Computer Engineering | Information Technology (for 2 Years upto 30-06-2019)

Affiliated to Maharashtra State Board of Technical Education (MSBTE) Mumbai

C E R T I F I C A T E

This is to certify that Mr. **Mohit Hemanth Shetty, Mr. Shaikh Mohammed Talha Salim, Mr. Mehta Parth Rahul** and **Mrs. Dave Maitry Nimesh** from **0522, Thakur Polytechnic** Institute having Enrollment No: **1805220458, 1805220463, 1805220428** and **1805220477** has completed project of final year having title 'An application for MSBTE affiliated colleges' during the academic year 2020-2021. The project completed in a group consisting of **4** persons under the guidance of Faculty Guide.

Name and Signature of Guide (Mrs. Vaishali Rane): ……

Telephone: 9869155605

---------- ----------

H.O.D (CO) **PRINCIPAL**

(Mrs. VAISHALI RANE) (Dr. S.M. GANECHARI)

2020-2021

A Project Report on

"AN APPLICATION FOR MSBTE AFFILIATED COLLEGES"

Submitted by

MOHIT HEMANTH SHETTY
SHAIKH MOHAMMED TALHA SALIM
MEHTA PARTH RAHUL
DAVE MAITRY NIMESH

Project Guide

Mrs. Vaishali Rane

DEPARTMENT OF COMPUTER ENGINEERING

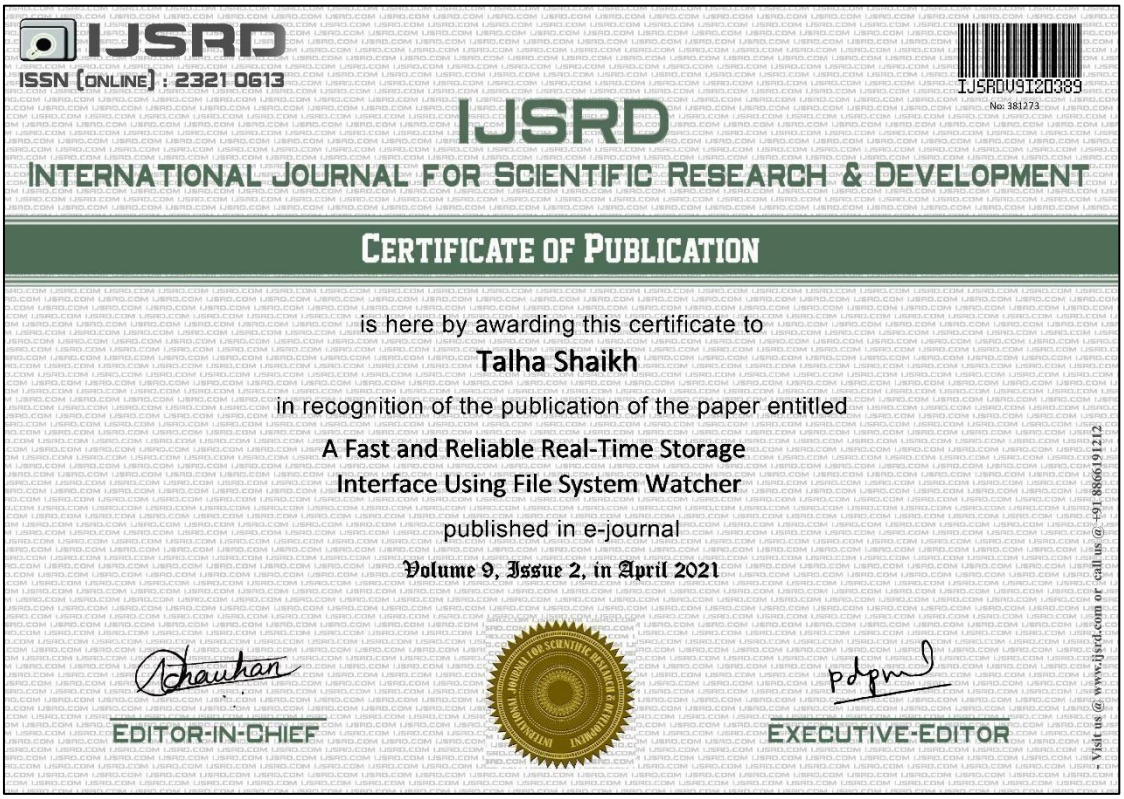
ZAGDU SINGH CHARITABLE TRUST (REGD.)

THAKUR POLYTECHNIC

(An ISO 9001:2015 Certified Institute**)** Thakur Complex, West to W.E. Highway, Kandivali (E), Mumbai – 400101.

2020-2021

Certificate of IJSRD for International Paper Publication:



IJSRD
ISSN (ONLINE) : 2321 0613
IJSRDV9I20389
No: 381272
IJSRD
INTERNATIONAL JOURNAL FOR SCIENTIFIC RESEARCH & DEVELOPMENT

CERTIFICATE OF PUBLICATION

is here by awarding this certificate to

Parth Mehta

in recognition of the publication of the paper entitled

A Fast and Reliable Real-Time Storage Interface Using File System Watcher

published in e-journal

Volume 9, Issue 2, in April 2021

EDITOR-IN-CHIEF

EXECUTIVE-EDITOR

IJSRD
ISSN (ONLINE) : 2321 0613
IJSRDV9I20389
No: 381274
IJSRD
INTERNATIONAL JOURNAL FOR SCIENTIFIC RESEARCH & DEVELOPMENT

CERTIFICATE OF PUBLICATION

is here by awarding this certificate to

Maitry Dave

in recognition of the publication of the paper entitled

A Fast and Reliable Real-Time Storage Interface Using File System Watcher

published in e-journal

Volume 9, Issue 2, in April 2021

EDITOR-IN-CHIEF

EXECUTIVE-EDITOR

National Level Certificates:

Akhil Bharatiya Maratha Shikshan Parishad's

Anantrao Pawar College of Engineering & Research,Pune

Accredited By NAAC , ISO 9001:2015 Certified
Under Mentorship Of College Of Engineering, Pune(COEP)
Sr. No.103,Shahu College Campus Parvati, Near Swargate ,Pune
College Code:6794

In Association with

Maharashtra State Board of Technical Education (MSBTE)

'APCOER TECHNOTHON -2021'

NATIONAL LEVEL ONLINE PROJECT, PAPER & POSTER COMPETITION FOR DIPLOMA STUDENTS

CERTIFICATE

This is to Certify that

Mr,/Ms **Maitry Dave** from Thakur Polytechnic, Kandivali Computer Engineering Department, has been awarded Ist prize in Group V in APCOER TECHNOTHON-2021 Online Project/Paper/Poster Competition held on 21st & 22nd May, 2021

Prof.A.N.Kalal	Prof. K.S.Jetha	Prof.Abhay Shelar	Dr. Sunil Thakare
Coordinator Info.Tech.Dept	HOD Info.Tech.Dept	Event Coordinator APCOER, PUNE	Principal APCOER, PUNE

Akhil Bharatiya Maratha Shikshan Parishad's

Anantrao Pawar College of Engineering & Research,Pune

Accredited By NAAC , ISO 9001:2015 Certified
Under Mentorship Of College Of Engineering, Pune(COEP)
Sr. No.103,Shahu College Campus Parvati, Near Swargate ,Pune
College Code:6794

In Association with

Maharashtra State Board of Technical Education (MSBTE)

'APCOER TECHNOTHON -2021'

NATIONAL LEVEL ONLINE PROJECT, PAPER & POSTER COMPETITION FOR DIPLOMA STUDENTS

CERTIFICATE

This is to Certify that

Mr,/Ms **Mohit Shetty** from Thakur Polytechnic, Kandivali Computer Engineering Department, has been awarded Ist prize in Group V in APCOER TECHNOTHON-2021 Online Project/Paper/Poster Competition held on 21st & 22nd May, 2021

Prof.A.N.Kalal	Prof. K.S.Jetha	Prof.Abhay Shelar	Dr. Sunil Thakare
Coordinator Info.Tech.Dept	HOD Info.Tech.Dept	Event Coordinator APCOER, PUNE	Principal APCOER, PUNE

Akhil Bharatiya Maratha Shikshan Parishad's

Anantrao Pawar College of Engineering & Research,Pune

Accredited By NAAC , ISO 9001:2015 Certified
Under Mentorship Of College Of Engineering, Pune(COEP)
Sr. No.103,Shahu College Campus Parvati, Near Swargate ,Pune
College Code:6794

In Association with

Maharashtra State Board of Technical Education (MSBTE)

'APCOER TECHNOTHON -2021'

NATIONAL LEVEL ONLINE PROJECT, PAPER & POSTER COMPETITION FOR DIPLOMA STUDENTS

CERTIFICATE

This is to Certify that

Mr,/Ms **Parth Mehta** from Thakur Polytechnic, Kandivali Computer Engineering Department, has been awarded Ist prize in Group V in APCOER TECHNOTHON-2021 Online Project/Paper/Poster Competition held on 21st & 22nd May, 2021

Prof.A.N.Kalal	Prof. K.S.Jetha	Prof.Abhay Shelar	Dr. Sunil Thakare
Coordinator Info.Tech.Dept	HOD Info.Tech.Dept	Event Coordinator APCOER, PUNE	Principal APCOER, PUNE

Akhil Bharatiya Maratha Shikshan Parishad's

Anantrao Pawar College of Engineering & Research,Pune

Accredited By NAAC , ISO 9001:2015 Certified
Under Mentorship Of College Of Engineering, Pune(COEP)
Sr. No.103,Shahu College Campus Parvati, Near Swargate ,Pune
College Code:6794

In Association with

Maharashtra State Board of Technical Education (MSBTE)

'APCOER TECHNOTHON -2021'

NATIONAL LEVEL ONLINE PROJECT, PAPER & POSTER COMPETITION FOR DIPLOMA STUDENTS

CERTIFICATE

This is to Certify that

Mr,/Ms **Talha Shaikh** from Thakur Polytechnic, Kandivali Computer Engineering Department, has been awarded Ist prize in Group V in APCOER TECHNOTHON-2021 Online Project/Paper/Poster Competition held on 21st & 22nd May, 2021

Prof.A.N.Kalal	Prof. K.S.Jetha	Prof.Abhay Shelar	Dr. Sunil Thakare
Coordinator Info.Tech.Dept	HOD Info.Tech.Dept	Event Coordinator APCOER, PUNE	Principal APCOER, PUNE

Acknowledgement

As a part of third year syllabus, we could successfully complete our project **"An application for MSBTE affiliated colleges"**. We feel immense pleasure in submitting the report, while submitting this report we avail this opportunity to express our gratitude toward all those who have guided and helped us in completing this task successfully. Heading the list is our honourable Principal **Dr. S.M. GANECHARI** who is beginner of our inspiration. We would like to thank our H.O. D of computer engineering **Mrs. VAISHALI RANE** for ardour in inciting the subject and her valuable suggestion.

We owe deep gratitude to our guide **Mrs. Vaishali Rane** who proved to be supportive guide to us. Apart from bringing to us what can be joy for creative, every time she acted promptly to correct our mistakes. The successful completion of this project is possible by her guidance and co-operation only, without which the work would have never been completed. We give our wholehearted thanks to our college **THAKUR POLYTECHNIC** for giving us the opportunity & support to the project. Finally, we wish to express our deep sense of respect and gratitude to our parents who always bear with us in any critical condition and to all others, for sparing their time and helping us for completion of this project in whatever way they could.

Sincere thanks to all group members.

Name of Members / Roll No

Mohit Hemanth Shetty [49]
Shaikh Mohammed Talha Salim [54]
Mehta Parth Rahul [20]
Dave Maitry Nimesh [68]

INDEX

Abstract

In today's world, teachers are generally expected to do a lot work except the task of just teaching or planning to teach. It could be related to creating lesson plan or any other form of documentation that is expected from the board or college or department or simply for the sake of record keeping. We as students have always observed these patterns and had thought of different ways to automate, if not all but at least some of those tasks, in order ease the lives of our teachers who take great efforts in order to ensure that keep learning even during the lockdown period from both their homes and colleges. There are other problems that are common to both students and teachers, so we have addressed one of those issues as well. So, our major project mainly tries to solve/automate three main problems/task. The first one being report generation. After the end of every semester, MSBTE releases the marksheets of all the students had previously successfully registered for the exams using the exam form. So, for the sake of record keeping teachers are expected to note down the marks from the marksheet of every student and every subject of the student which could be am eye-straining and time-consuming process. Our first component automates just that. Our second component basically automates the process of sharing the pending submissions for a given assignment in Google Classroom, so that teacher don't have to manually note the submission of a given assignment/all the assignment/selected assignments in on Google Classroom. This would make things a lot faster and save a lot of time of teachers during the ending of the semester. Our last component is basically a PHP based file system manager and IDE that specifically targets a few problems that we and our teachers faced while writing PHP scripts for practice and not only solves them but also makes the entire a lot more convenient and faster.

CHAPTER 1
INTRODUCTION

1.0 Introduction

The main aim of our project is to mainly solve all the problems teacher and students in MSBTE affiliated colleges face. So, based on that idea/approach, we have selected primarily three different problems and have developed software, application, and scripts for the same.

The first problem is basically that of report generation. At the end of every semester MSBTE releases the marksheets of all students who had successfully registered for their exams. For the sake of record keeping, teachers from all departments are expected to note the marks of each student of every year for that given summer/winter exam in the form of a gazette/excel sheet. The entire process of noting down all this data and presenting it in the form of excel could not only be very eye-straining and time consuming, but also wouldn't be guaranteed to be accurate as human errors can always occur and this eventually keeps the teachers in a lot of stress to keep re-checking the data before finally submitting it or storing it where it is expected to. So, to solve this problem we have basically created a GUI-based application in Java using the Swing GUI framework that we had learned in our 5^{th} semester under the subject of Advanced Java Programming. While developing the GUI, we also developed a library keeping the OOP nature of the language in mind (i.e. in the form of logical classes) in the same language, that can be used in the future by any student, who wants to make an application/project in Java by fetching marksheet data from MSBTE's server in order to create report(s) for each student or basically any part(s) of the process to fasten their process of development with the help of the library.

The second problem that we are trying to solve is that of noting the pending submissions in Google Classroom for a given (set of) assignment(s). When the semester comes to an end, all the submissions start to pour in, and the teacher is expected to assess them and return the assessment and the submissions itself to the concerned authority or specified repository in a short period of time. In between all the chaos, manually noting all the submissions that are pending can not only be very time consuming but also frustrating as the Google Classroom does not provide any simple way to either view or copy/share the students whose assignments are pending. It does a provide a complex table view displaying all those details, but that still makes the process of sharing the data by manually noting it, just to share it on another platform a lot more time consuming. So to solve this problem or to make things a lot more faster, we have developed an Android application (Java) using Firebase/GCP and the official Google Classroom API to fetch all the required classroom related details and design all the other required features or views accordingly. By using this app, the teacher could directly use his/her account to log into the application, grant the required permissions, select the classroom and choose the assignment (s)he would like to share based on all the options available on the screen (copy/share any one, copy all and search all copy selected).

The third problem that we are trying to solve here is basically that of being able to conveniently and quickly develop PHP scripts on a local server while adding additional options that are not currently supported by any IDE that is used to run PHP scripts/systems on the internet (e.g. controlling error reporting) to actually be able to control the environment. So, the third component solves a given set of problems that we had faced or things we felt could have been done faster otherwise. So, in order to support the IDE, we used an PHP file system along with it to create a complete independent system that can be used by a beginner to continently and quickly develop scripts.

1.1 Main Objectives of each component

1. MSBTE Report Generator (GUI + Library)

- It is used to generate a gazette for a given result link and a set of enrolment numbers and specified output file (.xls/.xlsx)
- It can be used as a library by another developer to build projects or applications based on the classes (s)he wants to re-use (for e.g. getting the details of a student in the form of an object which can be used to design report cards for each student, getting enrolment nos. from a different source other than the GUI and so on…)
- The component can work for different semesters and departments as well. So each unique type of entry will be on a separate sheet.

2. Pending Submissions Viewer

- It can be used to copy the status of all the students whose assignments haven't been checked.
- The application would also offer ways for the user to filter all the assignments and as well have an option to choose all the assignments to copy.

3. PHP File System and IDE

- Being able to conveniently and develop and debug applications that run directly on browsers.
- Custom error reporting options for PHP scripts.

CHAPTER-2

LITERATURE SURVEY

2.0. Web Scraping:

Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. The web scraping software may directly access the World Wide Web using the Hypertext Transfer Protocol or a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web crawler. It is a form of copying in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, for later retrieval or analysis.

Web scraping a web page involves fetching it and extracting from it. Fetching is the downloading of a page (which a browser does when a user views a page). Therefore, web crawling is a main component of web scraping, to fetch pages for later processing. Once fetched, then extraction can take place. The content of a page may be parsed, searched, reformatted, its data copied into a spreadsheet or loaded into a database. Web scrapers typically take something out of a page, to make use of it for another purpose somewhere else. An example would be to find and copy names and telephone numbers, or companies and their URLs, or e-mail addresses to a list (contact scraping).

Web scraping is used for contact scraping, and as a component of applications used for web indexing, web mining and data mining, online price change monitoring and price comparison, product review scraping (to watch the competition), gathering real estate listings, weather data monitoring, website change detection, research, tracking online presence and reputation, web mashup and, web data integration.

Web pages are built using text-based mark-up languages (HTML and XHTML), and frequently contain a wealth of useful data in text form. However, most web pages are designed for human endusers and not for ease of automated use. As a result, specialized tools and software have been developed to facilitate the scraping of web pages.

Newer forms of web scraping involve monitoring data feeds from web servers. For example, JSON is commonly used as a transport storage mechanism between the client and the web server.

There are methods that some websites use to prevent web scraping, such as detecting and disallowing bots from crawling (viewing) their pages. In response, there are web scraping systems that rely on using techniques in DOM parsing, computer vision and natural language processing to simulate human browsing to enable gathering web page content for offline parsing.

We have used web scraping in our report generator project to scrap information from the HTML page served by the official MSBTE server.

2.1 Firebase:

Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. In 2014, Google acquired the platform and it is now their flagship offering for app development.

Firebase evolved from Envolve, a prior startup founded by James Tamplin and Andrew Lee in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that were not chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it. They founded Firebase as a separate company in September 2011 and it launched to the public in April 2012.

Firebase's first product was the Firebase Realtime Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real-time, collaborative applications.

In May 2012, a month after the beta launch, Firebase raised $1.1 million in seed funding from venture capitalists Flybridge Capital Partners, Greylock Partners, Founder Collective, and New Enterprise Associates. In June 2013, the company further raised $5.6 million in Series A funding from Union Square Ventures and Flybridge Capital Partners.

In 2014, Firebase launched two products. Firebase Hosting and Firebase Authentication. This positioned the company as a mobile backend as a service.

In October 2014, Firebase was acquired by Google. A year later, in October 2015, Google acquired Divshot, an HTML5 web-hosting platform, to merge it with the Firebase team.

In May 2016, at Google I/O, the company's annual developer conference, Firebase introduced Firebase Analytics and announced that it was expanding its services to become a unified backendas-a-service (BaaS) platform for mobile developers. Firebase now integrates with various other Google services, including Google Cloud Platform, AdMob, and Google Ads to offer broader products and scale for developers. Google Cloud Messaging, the Google service to send push notifications to Android devices, was superseded by a Firebase product, Firebase Cloud Messaging, which added the functionality to deliver push notifications to both iOS and web devices. In January 2017, Google acquired Fabric and Crashlytics from Twitter to add those services to Firebase.

In October 2017, Firebase has launched Cloud Firestore, a real-time document database as the successor product to the original Firebase Realtime Database.

We have used this in our project to quickly setup google sign for our application without any major requirements for setting up a GCP project. We have used this in our Pending Submissions Viewer application.

2.2 Google Cloud Platform (GCP)

Google Cloud Platform (GCP), offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning. Registration requires a credit card or bank account details.

Google Cloud Platform provides infrastructure as a service, platform as a service, and serverless computing environments.

In April 2008, Google announced App Engine, a platform for developing and hosting web applications in Google-managed data centers, which was the first cloud computing service from the company. The service became generally available in November 2011. Since the announcement of App Engine, Google added multiple cloud services to the platform.

Google Cloud Platform is a part[6] of Google Cloud, which includes the Google Cloud Platform public cloud infrastructure, as well as Google Workspace (G Suite), enterprise versions of Android and Chrome OS, and application programming interfaces (APIs) for machine learning and enterprise mapping services.

Every Firebase project is a GCP project internally, so we have used it to enable the Classroom API for our application. We have used this in our Pending Submissions Viewer application.

2.3. Google Classroom API

Google Classroom is mission control for your classes. As a free service for teachers and students, you can create classes, distribute assignments, send feedback, and see everything in one place. Instant. Paperless. Easy.

The Google Classroom API gives developers access to their user's data and enables interoperable integrations.

Integration with Google Classroom is done through its API. The API gives developers a wide range of data that is used to increase interoperability between third-party content and Google Classroom.

We have used this in our Pending Submissions Viewer application.

2.4. Java Swing

Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT). Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

Unlike AWT components, Swing components are not implemented by platform-specific code. Instead, they are written entirely in Java and therefore are platform independent.

In December 2008, Sun Microsystems (Oracle's predecessor) released the CSS / FXML based framework that it intended to be the successor to Swing, called JavaFX.

The Internet Foundation Classes (IFC) were a graphics library for Java originally developed by Netscape Communications Corporation and first released on December 16, 1996. On April 2, 1997, Sun Microsystems and Netscape Communications Corporation announced their intention to incorporate IFC with other technologies to form the Java Foundation Classes. The "Java Foundation Classes" were later renamed "Swing."

Swing introduced a mechanism that allowed the look and feel of every component in an application to be altered without making substantial changes to the application code. The introduction of support for a pluggable look and feel allows Swing components to emulate the appearance of native components while still retaining the benefits of platform independence. Originally distributed as a separately downloadable library, Swing has been included as part of the Java Standard Edition since release 1.2. The Swing classes and components are contained in the javax.swing package hierarchy.

Development of Swing's successor, JavaFX, started in 2005, and it was officially introduced two years later at JavaOne 2007. JavaFX was open sourced in 2011 and, in 2012, it became part of the Oracle JDK download. JavaFX is replacing Swing owing to several advantages, including being more lightweight,

having CSS styling, sleek design controls, and the use of FXML and Scene Builder. In 2018, JavaFX was made a part of the OpenJDK under the OpenJFX project to increase the pace of its development.

Members of the Java Client team that was responsible for Swing included Jeff Dinkins (manager), Georges Saab, Jim Graham, Rick Levenson, Tim Prinzing, Jonni Kanerva, Jeannette Hung, Tom Ball, and Amy Fowler (technical lead). We have used this in our Report Generator application.

2.5. Android

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google. It was unveiled in November 2007, with the first commercial Android device, the HTC Dream, being launched in September 2008.

It is free and open-source software; its source code is known as Android Open Source Project (AOSP), which is primarily licensed under the Apache License. However most Android devices ship with additional proprietary software pre-installed, most notably Google Mobile Services (GMS) which includes core apps such as Google Chrome, the digital distribution platform Google Play and associated Google Play Services development platform.

About 70 percent of Android smartphones run Google's ecosystem; some with vendor-customized user interface and software suite, such as TouchWiz and later One UI by Samsung, and HTC Sense.[14] Competing Android ecosystems and forks include Fire OS (developed by Amazon) or LineageOS. However, the "Android" name and logo are trademarks of Google which impose standards to restrict "uncertified" devices outside their ecosystem to use Android branding.

The source code has been used to develop variants of Android on a range of other electronics, such as game consoles, digital cameras, portable media players, PCs and others, each with a specialized user interface. Some well known derivatives include Android TV for televisions and Wear OS for wearables, both developed by Google. Software packages on Android, which use the APK format, are generally distributed through proprietary application stores like Google Play Store, Samsung Galaxy Store, Huawei AppGallery, Cafe Bazaar, and GetJar, or open source platforms like Aptoide or F-Droid.

Android has been the best-selling OS worldwide on smartphones since 2011 and on tablets since 2013. As of May 2021, it has over three billion monthly active users, the largest installed base of any operating system, and as of January 2021, the Google Play Store features over 3 million apps. The current stable version is Android 11, released on September 8, 2020.

Android Inc. was founded in Palo Alto, California, in October 2003 by Andy Rubin, Rich Miner, Nick Sears, and Chris White. Rubin described the Android project as having "tremendous potential in developing smarter mobile devices that are more aware of its owner's location and preferences". The early intentions of the company were to develop an advanced operating system for digital cameras, and this was the basis of its pitch to investors in April 2004. The company then decided that the market for cameras was not large enough for its goals, and five months later it had diverted its efforts and was pitching Android as a handset operating system that would rival Symbian and Microsoft Windows Mobile.

Rubin had difficulty attracting investors early on, and Android was facing eviction from its office space. Steve Perlman, a close friend of Rubin, brought him $10,000 in cash in an envelope, and shortly thereafter wired an undisclosed amount as seed funding. Perlman refused a stake in the company and has stated "I did it because I believed in the thing, and I wanted to help Andy."

In July 2005, Google acquired Android Inc. for at least $50 million. Its key employees, including Rubin, Miner, Sears, and White, joined Google as part of the acquisition. Not much was known about the secretive Android Inc. at the time, with the company having provided few details other than that it was making software for mobile phones. At Google, the team led by Rubin developed a mobile device platform powered by the Linux kernel. Google marketed the platform to handset makers and carriers on the promise of providing a flexible, upgradeable system. Google had "lined up a series of hardware components and software partners and signaled to carriers that it was open to various degrees of cooperation".

Speculation about Google's intention to enter the mobile communications market continued to build through December 2006. An early prototype had a close resemblance to a BlackBerry phone, with no touchscreen and a physical QWERTY keyboard, but the arrival of 2007's Apple iPhone meant that Android "had to go back to the drawing board". Google later changed its Android specification documents to state that "Touchscreens will be supported", although "the Product was designed with the presence of discrete physical buttons as an assumption, therefore a touchscreen cannot completely replace physical buttons". By 2008, both Nokia and BlackBerry announced touch-based smartphones to rival the iPhone 3G, and Android's focus eventually switched to just touchscreens. The first commercially available smartphone running Android was the HTC Dream, also known as T-Mobile G1, announced on September 23, 2008.

On November 5, 2007, the Open Handset Alliance, a consortium of technology companies including Google, device manufacturers such as HTC, Motorola and Samsung, wireless carriers such as Sprint and T-Mobile, and chipset makers such as Qualcomm and Texas Instruments, unveiled itself, with a goal to develop "the first truly open and comprehensive platform for mobile devices".[33][34][35] Within a year, the Open Handset Alliance faced two other open source competitors, the Symbian Foundation and the LiMo Foundation, the latter also developing a Linuxbased mobile operating system like Google. In September 2007, InformationWeek covered an Evalueserve study reporting that Google had filed several patent applications in the area of mobile telephony.

Since 2008, Android has seen numerous updates which have incrementally improved the operating system, adding new features and fixing bugs in previous releases. Each major release is named in alphabetical order after a dessert or sugary treat, with the first few Android versions being called "Cupcake", "Donut", "Eclair", and "Froyo", in that order. During its announcement of Android KitKat in 2013, Google explained that "Since these devices make our lives so sweet, each Android version is named after a dessert", although a Google spokesperson told CNN in an interview that "It's kind of like an internal team thing, and we prefer to be a little bit — how should I say — a bit inscrutable in the matter, I'll say".

We have used this in our Pending Submissions Viewer application.

2.6. PHP

PHP is a general-purpose scripting language especially suited to web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page,[7] but it now stands for the recursive initialism PHP: Hypertext Pre-processor.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside of the web

context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification.

We used this to develop our PHP file system and IDE.

2.7. ace.js

Ace (from Ajax.org Cloud9 Editor) is a standalone code editor written in JavaScript. The goal is to create a web-based code editor that matches and extends the features, usability, and performance of existing native editors such as TextMate, Vim, or Eclipse. It can be easily embedded in any web page and JavaScript application. Ace is developed as the primary editor for Cloud9 IDE and as the successor of the Mozilla Skywriter project. We used this to develop our IDE (front-end).

Features:

- Syntax highlighting.
- Auto indentation and outdent.
- An optional command line.
- Work with large documents (handles hundreds of thousands of lines without issue).
- Fully customizable key bindings including vi and Emacs modes.
- Themes (TextMate themes can be imported).
- Search and replace with regular expressions.
- Highlight matching parentheses.
- Toggle between soft tabs and real tabs.
- Displays hidden characters.
- Highlight selected word.
- Multiple cursor selection.
- Column select and edit mode.

CHAPTER 3

SCOPE OF THE PROJECT

3.0. What is project scope?

Project scope is the part of project planning that involves determining and documenting a list of specific project goals, deliverables, tasks, costs and deadlines. The documentation of a project's scope, which is called a *scope statement* or *terms of reference*, explains the boundaries of the project, establishes responsibilities for each team member and sets up procedures for how completed work will be verified and approved.

During the project, this documentation helps the project team remain focused and on task. The scope statement also provides the team with guidelines for making decisions about change requests during the project. Note that a project's scope statement should not be confused with its charter; a project's charter simply documents that the project exists.

Large projects naturally tend to change as they progress. If a project has been effectively "scoped" at the beginning, then managing these changes will be easier. When documenting a project's scope, stakeholders should be as specific as possible to avoid scope creep, a situation in which one or more parts of a project end up requiring more work, time or effort because of poor planning or miscommunication.

3,0,1 PRODUCT SCOPE DESCRIPTION

The **product scope** is defined by the features and functions that characterize a **product**, service, or result. It is the result of the project or the solution to the problem. The **product** could be an entire new system, an enhancement to an existing system, or a defect fix

3.0.2 PROJECT ACCEPTANCES CRITERIA

Project Acceptance criteria are **criteria** that include performance **requirements** and essential conditions, which must be met before **project** deliverables are accepted (PMBOK® Guide). They set out the specific circumstances under which the user will accept the final output of the **project**.

3.0.3 PROJECT CONSTRAINTS

Project constraints are limiting factors for your **project** that can impact quality, delivery, and overall **project** success. Some say there are as many as 19 **project constraints** to consider, including resources, methodology, and customer satisfaction.

Since our project has mainly three components, we have described a separate project scope for each one of them.

3.1. REPORT GENERATOR PROJECT SCOPE STATEMENT

Component Name: REPORT GENERATOR (GUI and Library) DATE: 10/12/2020

PRODUCT SCOPE DESCRIPTION

Report Generator will generate report of the students whose enrolment number has been entered in the form of an excel sheet so that teachers don't have to enter the enrolment of every single student and get the marksheet and then covert it into excel format for further use hence making the task extremely easy and very fast and system is very reliable and works extremely well hence saving the previous time of teachers and student to get their marksheet

PROJECT DELIVARBLES

- Creating the reports in the form of excel sheet
- Can also be used for multiple semesters simultaneously
- Should be easy to use and setup
- Should create separate excel sheet in the same file when using multiple semester

PROJECT ACCEPTANCES CRITERIA

- It should show and retrieve the correct data and append in the appropriate place
- It should not have any overlapping if an enrolment number is entered twice
- It should be easy to use and setup

PROJECT CONSTRAINTS

- It should be a light application which is fast
- It should be able to retrieve all branch's marksheet • It should follow college excel storing format
- Time/resources to be spent on the project.

3.2. MY PHP SERVER PROJECT SCOPE STATEMENT

COMPONENT NAME: MyPHPServer DATE: 10/12/2020

PRODUCT SCOPE DESCRIPTION

My PHP Server is basically file manager which take use of PHP and MYSQL and also can be used as an IDE for executing PHP codes and HTML, CSS and JS it uses on tine compilation which allows you to see the changes in your website as soon as you save it and do some changes to it.

PROJECT DELIVARBLES

- IDE which can use and execute all the browser languages such as PHP, HTML, CSS, and JS.
- Should be able to all the task of a file manager such as delete, view, copy, etc.
- Should be able to show the changes on the website in real-time
- Should be lite weight and easy to use
- Should be able to show all data properly

PROJECT ACCEPTANCES CRITERIA

- It should show and retrieve the correct data and show properly
- It should not have any overlapping of data available
- It should be easy to use and setup
- Should be a compatible IDE
- Should have Realtime changes

PROJECT CONSTRAINTS

- It should be a light application which is fast
- It should be able to use all the web-based programming languages features
- Time/resources

3.3. PENDING SUBMISSIONS VIEWER PROJECT SCOPE STATEMENT

Component Name: Pending Submissions Viewer DATE: 10/12/2020

PRODUCT SCOPE DESCRIPTION

Pending Submissions Viewer will help us to see whose assignments are incomplete and whose are complete the traditional way is way too time consuming hence we have a method to save time and also the process is a lot fast compared to the traditional method it's an application which works in co-operation with google classroom like an addon.

PROJECT DELIVARBLES

- Should be lite weight and easy to use
- Should be able to show all data properly
- Should show results of all the subjects without any error

PROJECT ACCEPTANCES CRITERIA

- It Should show and retrieve the correct data and show properly
- It should not have any overlapping of data available.
- It should be easy to use and setup

PROJECT CONSTRAINTS

- It should be a light application which is fast
- It should be based on google classroom API.
- Time/resources

CHAPTER 4
DEVELOPMENT PHASE

4.0. SRS Document

4.0.1. What is an SRS Document?

A Software Requirements Specification (SRS) is a document that describes the nature of a project, software or application. In simple words, SRS document is a manual of a project provided it is prepared before you kick-start a project/application. This document is also known by the names SRS report, software document. A software document is primarily prepared for a project, software or any kind of application.
There are a set of guidelines to be followed while preparing the software requirement specification document. This includes the purpose, scope, functional and non-functional requirements, software and hardware requirements of the project. In addition to this, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project etc.

4.0.2 Index

4.0.3 Contents

1. Introduction

1.1 Purpose

1. Report Generator would generate a gazette for a given set of enrolment nos.

2. Pending Submissions Viewer shows us how many students are left to submit certain assignments

3. MyPHPServer would work as an IDE with a file manager supporting it.

1.2. Intended Audience and Reading Suggestions

This project is a prototype for the Report Generator, MyPHPServer and Pending Submissions Viewer system and it is restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the teachers and other staff members and as well as to the students.

1,3 Project Scope

Report Generator will generate report of the students whose enrolment number has been entered in the form of an excel sheet so that teachers don't have to enter the enrolment of every single student and get the marksheet and then covert it into excel format for further use hence making the task extremely easy and very fast and system is very reliable and works extremely well hence saving the previous time of teachers and student to get their marksheet

Pending Submissions Viewer will help us to see whose assignments are incomplete and whose are complete the traditional way is way too time consuming hence we have a method to save time and also the process is a lot fast compared to the traditional method it's an application which works in co-operation with google classroom kind like an addon.

My PHP Server is basically file manager which take use of PHP and MYSQL and also can be used as an IDE for executing PHP codes and HTML, CSS and JS it uses on tine compilation which allows you to see the changes in your website as soon as you save it and do some changes to it.

1.4 References

https://www.kopykitab.com/Advanced-Java-Programming-by-Mahesh-Gurunani-Rajesh-YemulParvez-Vaghela

https://www.kopykitab.com/Web-Based-Application-Development-WITH-php-For-MSBTE-by-AA-Puntambekar

https://www.kopykitab.com/Web-Based-Application-Development-WITH-php-For-MSBTE-by-AA-Puntambekar

https://github.com/alexantr/filemanager

https://github.com/prasathmani/tinyfilemanager https://developers.google.com/classroom

https://developers.google.com/classroom/guides/auth

https://developers.google.com/classroom/quickstart/java

2. Overall Description

2.1 Product Perspective

Our CPE project contains these 3 modules

1. Report Generator would generate a gazette for a given set of enrolment nos.
2. Pending Submissions Viewer shows us how many students are left to submit certain assignments
3. MyPHPServer would work as an IDE with a file manager supporting it.

2.2. Product Features

The major features of our major project are:

Report Generator able to acquire all the marksheet

My PHP server has many features like error reporting, IDE, file manager.

Pending Submissions Viewer shortens the work of writing each student's name down instead gives a list automatically

2.3. Operating Environment

Operating system: Windows.
Platform: Android Studio, VS Code
Services- Firebase, GCP, Google Classroom API

2.4. User Characteristics

It is considered that the user does have the basic knowledge of operating the internet and to have access to it.
The administrator is expected to be familiar with the interface of the tech support system.

3. System Features

3.1 Functional Requirements

Operating system: Windows.
Platform: Android Studio, VS Code

Services- Firebase, GCP, Google Classroom API

4. External Interface Requirements

4.1 User Interfaces

- Front-end software: VS Code
- Back-end software: XAMPP

4.2 Hardware Interfaces

- Windows
- A browser which supports CGI, HTML & JavaScript

4.1. SQA Activities

The SEI (Software Engineering Institute) recommends a set of SQA activities that address quality assurance planning, record keeping, analysis, and reporting. Various SQA activities, which are performed by independent SQA group.

Objectives of Audit:

The aim of a conducting software audit is to provide an independent evaluation of the software products and processes to applicable standards, guidelines, plans, and procedures against compliance.

Objectives of Audit:

The aim of a conducting software audit is to provide an independent evaluation of the software products and processes to applicable standards, guidelines, plans, and procedures against compliance.

Roles and Responsibilities of Formal Audit:

• Manager: The manager decides on what needs to be reviewed and ensures that there is sufficient time allocated in the project plan for all of the required review activities. Managers do not usually get involved in the actual review process.

• Moderator: The Moderator, also known as lead reviewer, reviews the set of documents. The moderator will make the final decision as whether or NOT to release an updated document.

• Author: The author is the writer, who develops the document(s) to be reviewed. The author also takes responsibility for fixing any agreed defects.

• Scribe/Recorder: The scribe attends the review meeting and documents all of the issues/defect/problems and open points that were identified during the meeting.

During inspection the documents are prepared and checked thoroughly by the reviewers before the meeting. It involves peers to examine the product. A separate preparation is carried out during which the product is examined, and the defects are found. The defects found are documented in a logging list or issue log

The goals of inspection are:

1. It helps the author to improve the quality of the document under inspection

2. It removes defects efficiently and as early as possible

3. It improves product quality
4. It creates common understanding by exchanging information
5. It learns from defects found and prevent the occurrence of similar defects

Audits: We talked and about the project and searched for any logical or normal error as in audit we self-assessed ourselves.

Inspection-in inspection we group discussed about the project and searched for error and reconfirmed our timeline i.e.-are we on schedule and tested the modules which were created

4.2. Testing of our components

4.2.1 Black Box Testing

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing. Specific knowledge of the application's code, internal structure and programming knowledge in general is not required. The tester is aware of *what* the software is supposed to do but is not aware of *how* it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of *how* the software produces the output in the first place

We tested our project modules against the expected outputs and we got some error and we solved them and wrote them in defect report. We tested modules fuctions like retrieving data and appending process and if there is overlapping etc

Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding. The primary source of black box testing is a specification of requirements that is stated by the customer.

In this method, tester selects a function and gives input value to examine its functionality, and checks whether the function is giving expected output or not. If the function produces correct output, then it is passed in testing, otherwise failed. The test team reports the result to the development team and then tests the next function. After completing testing of all functions if there are severe problems, then it is given back to the development team for correction.

Generic steps of black box testing

- The black box test is based on the specification of requirements, so it is examined in the beginning.
- In the second step, the tester creates a positive test scenario and an adverse test scenario by selecting valid and invalid input values to check that the software is processing them correctly or incorrectly.
- In the third step, the tester develops various test cases such as decision table, all pairs test, equivalent division, error estimation, cause-effect graph, etc.
- The fourth phase includes the execution of all test cases.

- In the fifth step, the tester compares the expected output against the actual output
- In the sixth and final step, if there is any flaw in the software, then it is cured and tested again.

4.2.2 White Box Testing

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing White-box testing is a method of testing the application at the level of the source code. These test cases are derived through the use of the design techniques mentioned above: control flow testing, data flow testing, branch testing, path testing, statement coverage and decision coverage as well as modified condition/decision coverage. White-box testing is the use of these techniques as guidelines to create an error-free environment by examining all code. These white-box testing techniques are the building blocks of white-box testing, whose essence is the careful testing of the application at the source code level to reduce hidden errors later on.These different techniques exercise every visible path of the source code to minimize errors and create an error-free environment. The whole point of white-box testing is the ability to know which line of the code is being executed and being able to identify what the correct output should be

The box testing approach of software testing consists of black box testing and white box testing. We are discussing here white box testing which also known as glass box is **testing, structural testing, clear box testing, open box testing and transparent box testing**. It tests internal coding and infrastructure of a software focus on checking of predefined inputs against expected and desired outputs. It is based on inner workings of an application and revolves around internal structure testing. In this type of testing programming skills are required to design test cases. The primary goal of white box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

The term 'white box' is used because of the internal perspective of the system. The clear box or white box or transparent box name denote the ability to see through the software's outer shell into its inner workings.

Developers do white box testing. In this, the developer will test every line of the code of the program. The developers perform the White-box testing and then send the application or the software to the testing team, where they will perform the black box testing and verify the application along with the requirements and identify the bugs and sends it to the developer.

The developer fixes the bugs and does one round of white box testing and sends it to the testing team. Here, fixing the bugs implies that the bug is deleted, and the particular feature is working fine on the application.

Here, the test engineers will not include in fixing the defects for the following reasons:

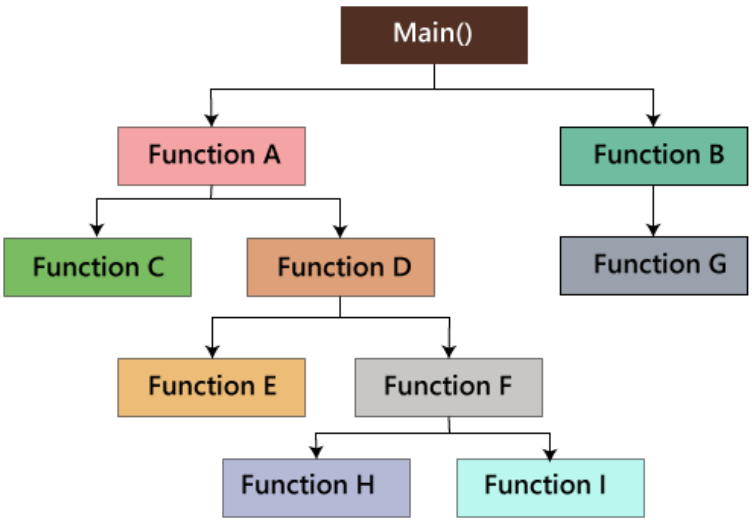
- Fixing the bug might interrupt the other features. Therefore, the test engineer should always find the bugs, and developers should still be doing the bug fixes.
- If the test engineers spend most of the time fixing the defects, then they may be unable to find the other bugs in the application.

The white box testing contains various tests, which are as follows:

- Path testing
- Loop testing
- Condition testing

Path testing

In the path testing, we will write the flow graphs and test all independent paths. Here writing the flow graph implies that flow graphs are representing the flow of the program and also show how every program is added with one another as we can see in the below image:



And test all the independent paths implies that suppose a path from main() to function G, first set the parameters and test if the program is correct in that particular path, and in the same way test all other paths and fix the bugs.

We used the testing method of basic path testing which is one of the type of Whitebox testing which helps us determine if the logic used in our program is correct or incorrect and also uncovers logical problem's as our project isn't very big we didn't use any automation tools to perform the white box testing.

4.2.3 Unit testing

In computer programming, unit testing is a software testing method by which individual units of source code—sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures—are tested to determine whether they are fit for use. Unit tests are typically automated tests written and run by software developers to ensure that a section of an application (known as the "unit") meets its design and behaves as intended. In procedural programming, a unit could be

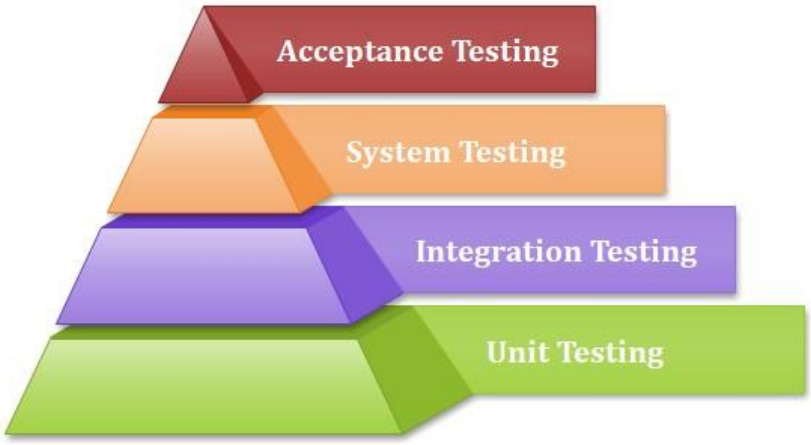
an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, or an individual method. By writing tests first for the smallest testable units, then the compound behaviours between those, one can build up comprehensive tests for complex applications

The purpose of unit testing is to test the correctness of isolated code. A unit component is an individual function or code of the application. White box testing approach used for unit testing and usually done by the developers.

Whenever the application is ready and given to the Test engineer, he/she will start checking every component of the module or module of the application independently or one by one, and this process is known as **Unit testing** or **components testing**.

Why Unit Testing?

In a testing level hierarchy, unit testing is the first level of testing done before integration and other remaining levels of the testing. It uses modules for the testing process which reduces the dependency of waiting for Unit testing frameworks, stubs, drivers and mock objects are used for assistance in unit testing.



Generally, **the** software goes under four level of testing: Unit Testing, Integration Testing, System Testing, and Acceptance Testing but sometimes due to time consumption software testers does minimal unit testing but skipping of unit testing may lead to higher defects during Integration Testing, System Testing, and Acceptance Testing or even during Beta Testing which takes place after the completion of software application.

Some crucial reasons are listed below:

- Unit testing helps tester and developers to understand the base of code that makes them able to change defect causing code quickly.
- Unit testing helps in the documentation. ○ Unit testing fixes defects very early in the development phase that's why there is a possibility to occur a smaller number of defects in upcoming testing levels.

○ It helps with code reusability by migrating code and test cases.

We used method of the unit testing while doing unit testing it is type of white box testing which means the tester should know the inner working of the code we tested each small module of the respective modules

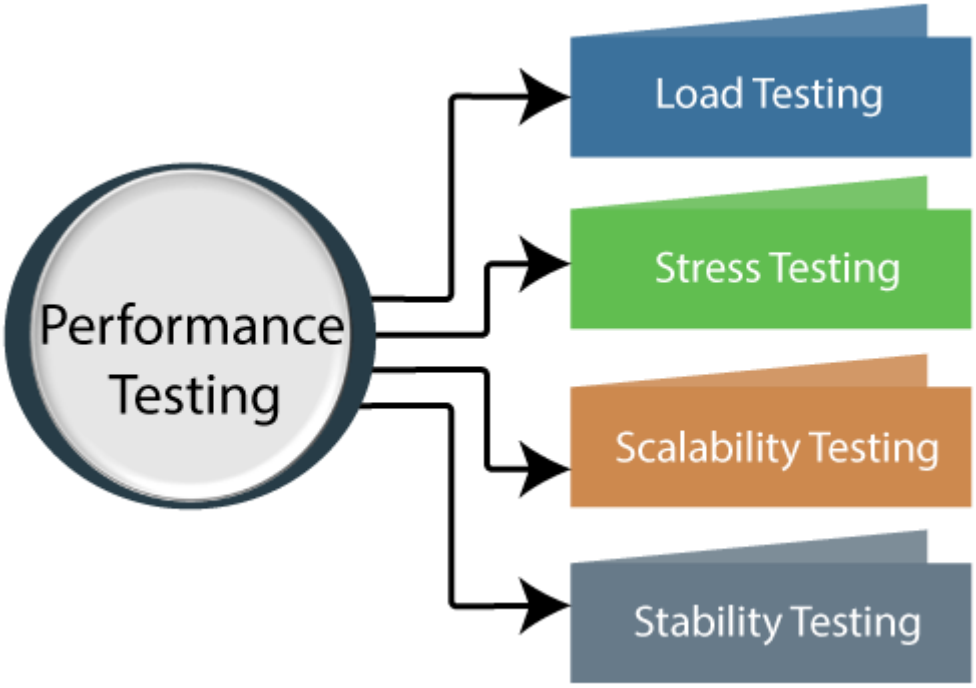
and then created one single module our report generator module was made in java language and classroom api in android studio and Myphp using PHP HTML and javascript so we tested each small module of the main module and made sure that each module works without any issues as the main motive of unit testing is that every module should be able to work alone

4.2.4 Performance testing

Performance Testing is a type of software testing that ensures software applications to perform properly under their expected workload. ... It is a testing method performed to determine the system performance in terms of speed, reliability and stability under varying workload.

The focus of Performance Testing is checking a software program's

- Speed - Determines whether the application responds quickly
- Scalability - Determines maximum user load the software application can handle. • Stability - Determines if the application is stable under varying loads • It is the most important part of non-functional testing.
- *Checking the behaviour of an application by applying some load is known as performance testing.*
- Generally, this testing defines how quickly the server responds to the user's request.
- While doing performance testing on the application, we will concentrate on the various factors like **Response time, Load, and Stability** of the application.
- **Response time:** Response time is the time taken by the server to respond to the client's request.
- **Load:** Here, Load means that when **N-number** of users using the application simultaneously or sending the request to the server at a time.
- **Stability:** For the stability factor, we can say that, when N-number of users using the application simultaneously for a particular time.



We also conducted performance testing to make sure that all our modules work with recommended workload we tested it by giving them 160 students enrolment number 160 is the average number of students in a branch but the report generator can even take more number of students

In the classroom API we retrieved the data of a whole classroom and it worked extremely well and appended the data as expected

In MYPHP server it showed the result in real time when we saved the code and all the rest of the function also worked well

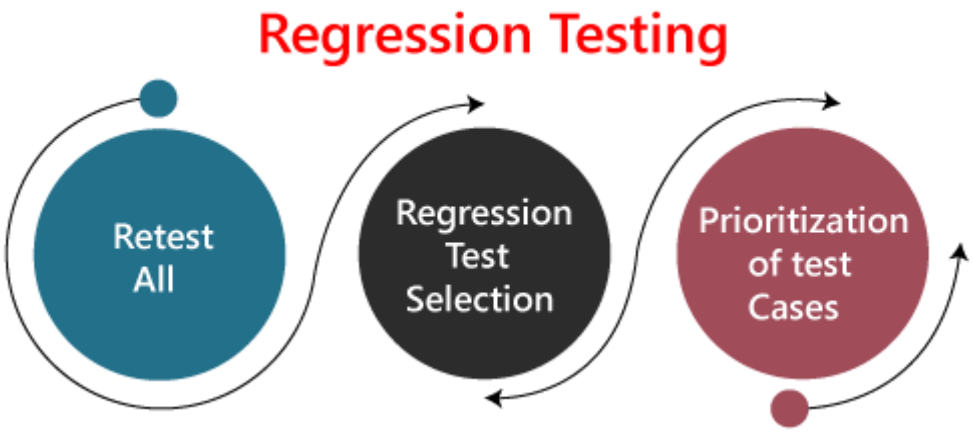
4.2.5 Regression Testing

Regression testing is a black box testing technique. It is used to authenticate a code change in the software does not impact the existing functionality of the product. Regression testing is making sure that the product works fine with new functionality, bug fixes, or any change in the existing feature.

Regression testing is a type of software testing. Test cases are re-executed to check the previous functionality of the application is working fine, and the new changes have not produced any bugs.

Regression testing can be performed on a new build when there is a significant change in the original functionality. It ensures that the code still works even when the changes are occurring. Regression means Re-test those parts of the application, which are unchanged.

Regression tests are also known as the Verification Method. Test cases are often automated. Test cases are required to execute many times and running the same test case again and again manually, is time-consuming and tedious too.

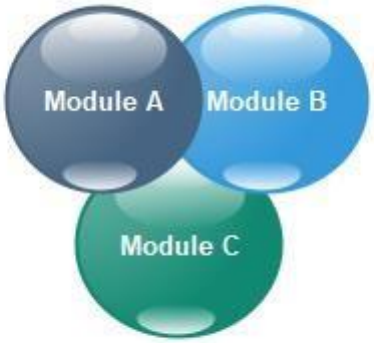


We also did regression testing to insure even if there was a change in the code made the logic did not change

4.2.6 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before validation testing.

Tested in Unit Testing



Under Integration Testing

Once all the components or modules are working independently, then we need to check the data flow between the dependent modules is known as **integration testing**.

After unit testing we needed to make sure that all the unit of the modules worked with each other and did what was expected and they all worked fine and they were no error

4.2.7 Stress Testing

In this section, we are going to understand **Stress Testing**, which is an important part of **Performance testing** and used to checks the behavior of an application by applying a load greater than the desired load.

Acceptance testing

Acceptance testing is formal testing based on user requirements and function processing. It determines whether the software is conforming specified requirements and user requirements or not. It is conducted as a kind of Black Box testing where the number of required users involved testing the acceptance level of the system. It is the fourth and last level of software testing.

User acceptance testing (UAT) is a type of testing, which is done by the customer before accepting the final product. Generally, UAT is done by the customer (domain expert) for their satisfaction, and check whether the application is working according to given business scenarios, real-time scenarios.

In this, we concentrate only on those features and scenarios which are regularly used by the customer or mostly user scenarios for the business or those scenarios which are used daily by the end-user or the customer.

4.2.8 Test cases

4.2.8.1 Report Generator

TC ID	Specifications	STEPS	Input Data	Expected Output	Actual Output	Status
TC_01	Slider	Run the Report Generator using ReportGenerator.vbs file	The ReportGenerator.vbs should be available and clickable.	The ReportGenerator.vbs file should be clicked	The ReportGenerator.vbs file is clicked.	Pass
TC_02	Slider	Run the Report Generator using run.bat file	The run.bat should be available and clickable.	The run.bat file should be clicked	The run.bat file is clicked.	Pass
TC_03	Continue Button	Click on the continue button	The continue button should be available and continue.	The continue button should be clicked.	The continue button is clicked.	Pass
TC_04	Sample/Example Link TextBox	Click on the Enter a Sample/ Example Link TextBox	Enter the Sample Link of MSBTE Result.	The Sample Link should be entered.	The Sample Link is Entered.	Pass

TC_05	Enter a list of UIDs TextArea	Click on the Enter a List of UIDs TextArea	Enter the List of UIDs(Enrollment/Seat No.s)	The List of UIDs should be entered.	The List of UIDs is Entered.	Pass
TC_06	Enter the name of output file TextBox	Click on the Enter the name of the output file TextBox	Enter the location of Output File Location	The Output File Location should be set.	The Output File Location is not set.	Fail
TC_07	Browse Button	Click on the Browse Button	The Browse Button should be clickable and available.	When clicking on the Browse Option a dialog box to Choose the destination folder/file should open.	When clicked on the Browse Option a dialog box to Choose the destination folder/file is opened.	Pass
TC_08	Generate Button	Click on Generate Button	The Generate Button should be available and Clickable.	Generating Report Dialog Box Should be opened and Report Generation should be started.	Generating Report Dialog Box is opened and Report Generation started.	Pass
TC_09	Invalid UIDs TextArea	The Invalid UIDs TextArea should be visible	The Invalid UIDs TextArea should be available.	The Invalid UIDs from the entered UIDs should be visible there.	The Invalid UIDs from the entered UIDs are not visible there.	Fail
TC_10	Generated Report	The Report Should be Generated in Excel Sheet	The Excel Sheet should be visisble.	The Data should be properly Generated in an Excel Sheet.	The Data is properly Generated in an Excel Sheet.	Pass

4.2.8.2 MyPHPServer

TC ID	Specifications	STEPS	Input Data	Expected Output	Actual Output	Status
TC_1	DELETE	RUN THE PROGRAM AND PRESS THE DELETE ACTION	NONE	DELETION OF FILE	DELETION OF FILE IS SUCCESSFUL	PASS
TC_2	RENAME	RUN THE PROGRAM AND PRESS THE RENAME ACTION	NONE	RENAME OPTION OF FILE	RENAME OPTION OF FILE IS SUCCESSFUL	PASS
TC_3	PREVIEW	RUN THE PROGRAM AND PRESS THE PREVIEW ACTION	NONE	PREVIEWING OF THE FILE	PREVIEWING OF THE FILE IS SUCCESFUL	PASS
TC_4	COPY	RUN THE PROGRAM AND PRESS THE COPY ACTION	NONE	COPY TO CLIPBOARD	COPY TO CLIPBOARD SUCCESFULLY	PASS
TC_5	DIRECT LINK	RUN THE PROGRAM	NONE	DIRECTLINK MAKING	DIRECT LINK IS MADE	PASS
TC_6	DOWNLOAD	RUN THE PROGRAM AND PRESS THE DOWNLOAD ACTION	NONE	DONLOADING OF THE FILE	DOWNLOADING OF THE FILE IS UNSUCCESSFUL FILE NOT DOWNLOADED	FAIL
TC_7	ERROR SORTING	RUN THE PROGRAM AND PRESS ERROR SORTING	NONE	ERROR SORTED	ERROR ARE SORTED	PASS
TC_8	REALTIME CHANGE	RUN THE PROGRAM AND RUN CODE	NONE	REALTIME CHANGES IN A FILE WHEN SAVED	REALTIME CHANGES IN A FILE WHEN SAVED IS SUCCESSFUL	PASS
TC_9	SUPPORTING ALL LANGUAGES	RUN THE PROGRAM AND RUN CODE	RUN PHP, HTML, CSS, JAVASCRIPT PROGRMS	SUPPORT TO ALL LANGUAGES	SUPPORT TO ALL LANGUAGES IS THERE	PASS

TC_10	SEARCH	RUN THE PROGRAM AND PRESS THE SEARCH ACTION	NONE	SEARCH OPTION WORKING	SEARCH OPTION IS WORKING	PASS
TC_11	SELECT	RUN THE PROGRAM AND PRESS THE SELECT ACTION	NONE	SELECT OPTION WORKING	SELECT OPTION IS WORKING AND CAN MULTI SELECT TOO	PASS
TC_12	UPLOAD	RUN THE PROGRAM AND PRESS THE UPLOAD ACTION	NONE	UPLOAD OPTION WORKING	UPLOAD OPTION IS WORKING	PASS
TC_13	ZIP	RUN THE PROGRAM AND PRESS THE ZIP ACTION	NONE	ZIP IS MADE OF FILE	ZIP IS NOT MADE OF FILE ERROW IN IMPLEMENTING LIBRARY	FAIL
TC_14	TAR	RUN THE PROGRAM AND PRESS THE TAR ACTION	NONE	TAR IS MADE OF FILE		FAIL
TC_15	FOLDER	RUN THE PROGRAM	NONE	ALL FOLDERS ARE VISIBLE AND ARE PROPER FORMAT	TAR IS NOT MADE OF FILE ERROR IN IMPLEMENTING LIBRARY	PASS
TC_16	COLORFULL TEXT	RUN THE PROGRAM AND OPEN SOME FILE LIKE TXT OR PHP OR HTML		COLORFULL WORDS IN EDITING VIEW	COLORFULL WORDS ARE THERE IN EDITING VIEW	PASS

4.2.8.3 Pending Submissions Viewer

TC ID	Specifications	STEPS	Input Data	Expected Output	Actual Output	Status
TC_1	CLASSROOM CONNECTION	START THE API	NONE	GOOGLE CLASSROOM SHOULD WORK WITH OUR API	GOOGLE CLASSROOM WORKS WITH OUR API	PASS

TC_2	CLASSROOM ACCESS	START THE API	NONE	CLASSROOM SHOULD GIVE ACCESS TO VIEW CLASS ROSTER ETC	CLASSROOM GIVES ACCESS TO VIEW CLASS ROSTER ETC	PASS
TC_3	CLASSROOM LOGIN	START THE API AND START THE LOGIN PROCESS	LOGIN INFO	SHOULD BE ABLE TO USE YOUR GID	IS ABLE TO USE YOUR GID	PASS
TC_4	RETRIVING OF DATA	START THE API AFTER LOGIN CLICK ON RETRIVE OPTION	NONE	DATA SHOULD BE RETRIVED FROM CLASSROOM REGARDING STUDENTS	DATA IS BE RETRIVED FROM CLASSROOM REGARDING STUDENTS	PASS
TC_5	COPYING TO CLIPBOARD	START THE API AFTER LOGIN CLICK ON RETRIVE OPTION	NONE	COPY TO CLIP BOARD FUCTION SHOULD WORK	COPY TO CLIP BOARD FUCTION WORKS	PASS
TC_6	CLASSROOM WITH NO STUDENT	START THE API AFTER LOGIN CLICK ON COPY TO CLIPBOARD OPTION	NONE	CLASROOM WITH NO STUDENTS SHOULD NOT SHOW ANYTHING AND SHOULD GIVE EXCEPTION	CLASROOM WITH NO STUDENTS NOT SHOWS ANYTHING AND SHOULD GIVE EXCEPTION	FAIL
TC_7	OPTION 1	START THE API AFTER LOGIN CLICK ON COPY TO OPTION 1	NONE	OPTION 1 SHOULD BE SHOWN IN COPY CLIPBOARD OPTION	OPTION 1 IS SHOWN IN COPY CLIPBOARD OPTION	PASS
TC_8	OPTION 2 RETURNED	START THE API AFTER LOGIN CLICK ON	NONE	RETURNED OPTION SHOULD BE SHOWN IN THE RETRIVED DATA	RETURNED OPTION IS SHOWN IN THE RETRIVED DATA	PASS
		COPY TO OPTION 2				
TC_9	OPTION 3 CREATE	START THE API AFTER LOGIN CLICK ON COPY TO OPTION 3	NONE	CREATION OPTION SHOULD WORK SHOULD SHO CREATION IN COPY TO CLIPBOARD	CREATION OPTION WORKS SHOULD SHO CREATION IN COPY TO CLIPBOARD	PASS

4.2.9 Defect Reports

The following defect reports were generated during the development phase of our project:

Defect Report 1

ID	1
PROJECT	MSBTE affiliated college program
PRODUCT	MSBTE affiliated college program Report generator
RELEASE VERSION	1.0.0
MODULE	Output File Location
DETECTED BUILD VERSION	1.0.0.1
SUMMARY	The main defect is in Output File Location.
DESCRIPTION	The value of text box is not getting stored in declared variable.
STEPS TO REPLICATE	1. Again, we have to check the variable declaration code section. 2. We have to change variable declaration code initialization.
ACTUAL RESULTS	when user click on output file textbox the location is not set.
EXPECTED RESULTS	when user click on output file textbox the location is set.
DEFECT SEVERITY	Medium
DEFECT PRIORITY	Medium
REPORTED BY	MAITRY D
ASSIGNED TO	TALHA S
STATUS	FIXED
FIXED BUILD VERSION	1.0.0.2

Defect Report 2

ID	**2**
PROJECT	MSBTE Report Generator
PRODUCT	MSBTE Report Generator
RELEASE VERSION	2.0.0
MODULE	UIDs Text Area
DETECTED BUILD VERSION	2.0.0.1
SUMMARY	The main defect is in UIDs Text Area.
DESCRIPTION	The value of Text Area is not getting stored in declared variable.
STEPS TO REPLICATE	1. Again we have to check the variable declaration in code section. 2.we have to change variable declaration code initialization.
ACTUAL RESULTS	The Invalid UIDs from the entered UIDs are not visible there.

EXPECTED RESULTS	The Invalid UIDs from the entered UIDs should be visible there.
DEFECT SEVERITY	High
DEFECT PRIORITY	High
REPORTED BY	PARTH M
ASSIGNED TO	MOHIT S
STATUS	FIXED
FIXED BUILD VERSION	2.0.0.2

Defect Report 3

ID	**3**
PROJECT	MyPHPServer
PRODUCT	MyPHPServer
RELEASE VERSION	3.0.0
MODULE	DOWNLOAD ACTION.
DETECTED BUILD VERSION	3.0.0.1
SUMMARY	The main defect is in download action.
DESCRIPTION	When we press the download, the download of the file is not successful.
STEPS TO REPLICATE	1. Again we must check the program code 2.we have to change program code
ACTUAL RESULTS	When the user hits download the file must be downloaded.
EXPECTED RESULTS	When the user press downloads the file is not downloading
DEFECT SEVERITY	Medium
DEFECT PRIORITY	Medium
REPORTED BY	TALHA S
ASSIGNED TO	MOHIT S
STATUS	FIXED
FIXED BUILD VERSION	3.0.0.1

Defect Report 4

ID	**4**
PROJECT	MSBTE affiliated college program
PRODUCT	MSBTE affiliated college program PHP Program
RELEASE VERSION	4.0.0
MODULE	ZIP
DETECTED BUILD VERSION	4.0.0.1

SUMMARY	The main defect is in ZIP
DESCRIPTION	The ZIP is not made of file error in implementing library
STEPS TO REPLICATE	RUN THE PROGRAM AND PRESS THE ZIP ACTION
ACTUAL RESULTS	ZIP IS NOT MADE OF THE FILE
EXPECTED RESULTS	ZIP IS MADE OF THE FILE
DEFECT SEVERITY	SEVERE
DEFECT PRIORITY	HIGH
REPORTED BY	MOHIT S
ASSIGNED TO	MAITRY D
STATUS	FIXED
FIXED BUILD VERSION	4.0.0.2

Defect Report 5

ID	**5**
PROJECT	MSBTE Report Generator
PRODUCT	MSBTE Report Generator
RELEASE VERSION	5.0.0
MODULE	TAR
DETECTED BUILD VERSION	5.0.0.1
SUMMARY	The main defect is in TAR
DESCRIPTION	The TAR is not made of file error in implementing library
STEPS TO REPLICATE	RUN THE PROGRAM AND PRESS THE TAR ACTION
ACTUAL RESULTS	TAR IS NOT MADE OF THE FILE
EXPECTED RESULTS	TAR IS MADE OF THE FILE
DEFECT SEVERITY	HIGH
DEFECT PRIORITY	HIGH
REPORTED BY	TALHA S
ASSIGNED TO	PARTH M
STATUS	FIXED
FIXED BUILD VERSION	5.0.0.2

CHAPTER 5
TIMELINE CHART

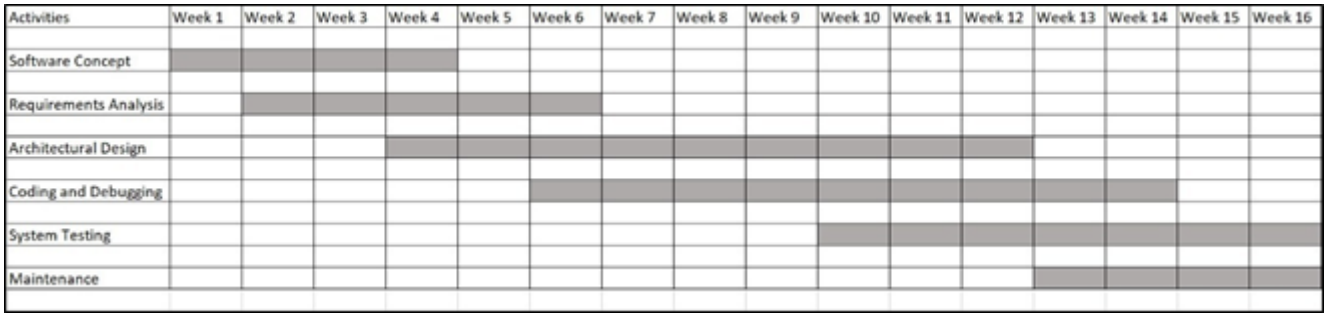


Fig. 5 Timeline chart

CHAPTER 6

OUTPUT OF MAJOR PROJECT

6.1. MSBTE Report Generator

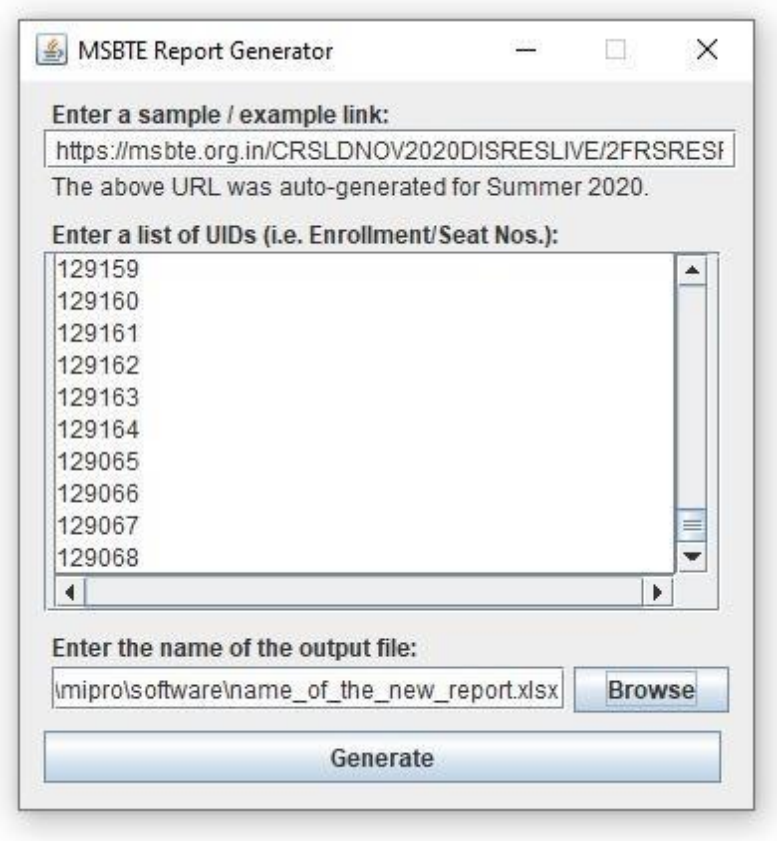


Fig. 6.1.1 A screenshot of the main report generator view

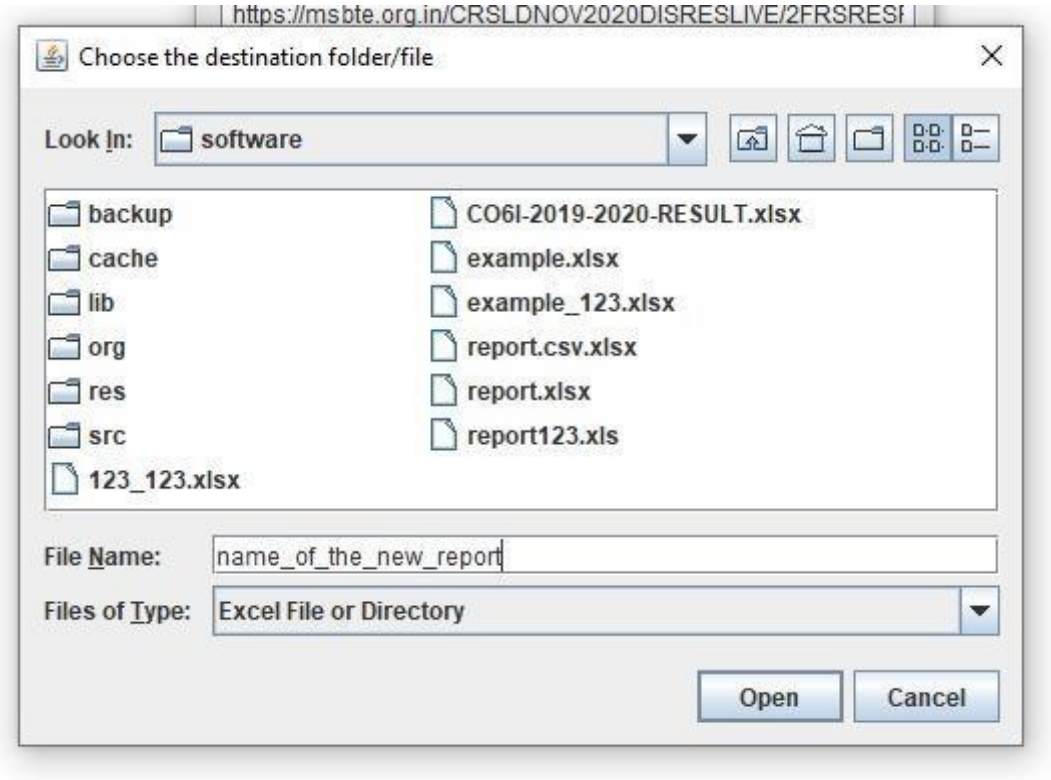


Fig. 6.1.1 A screenshot of the main report generator view

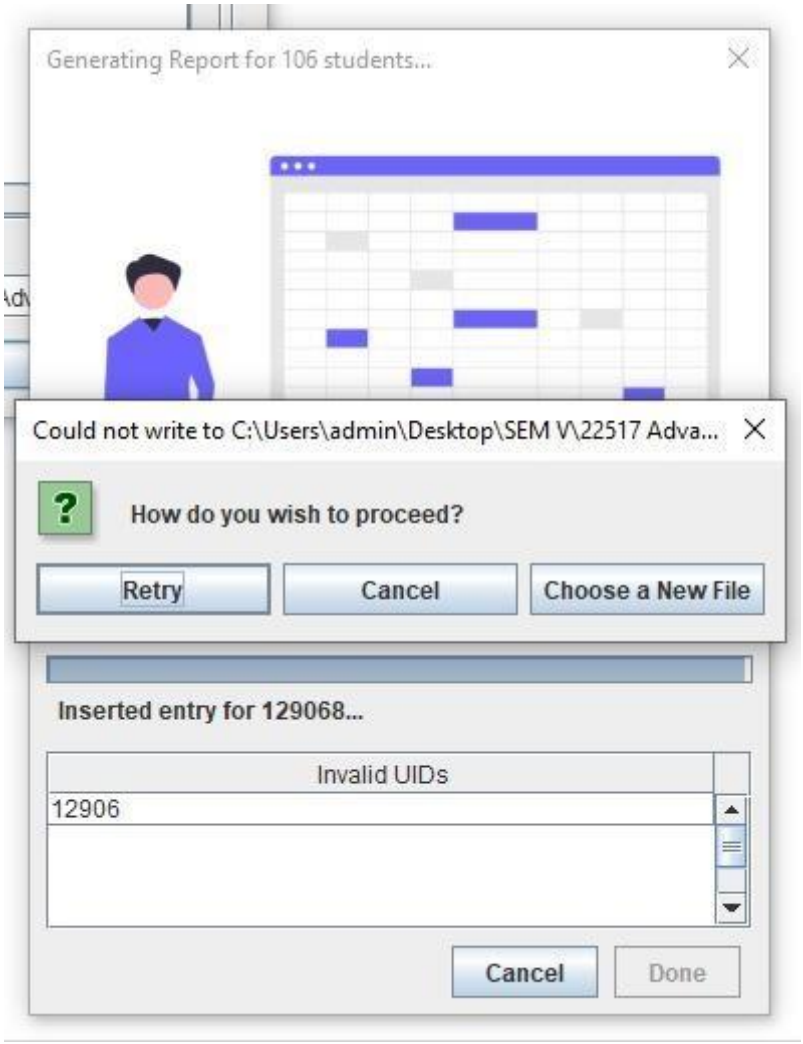
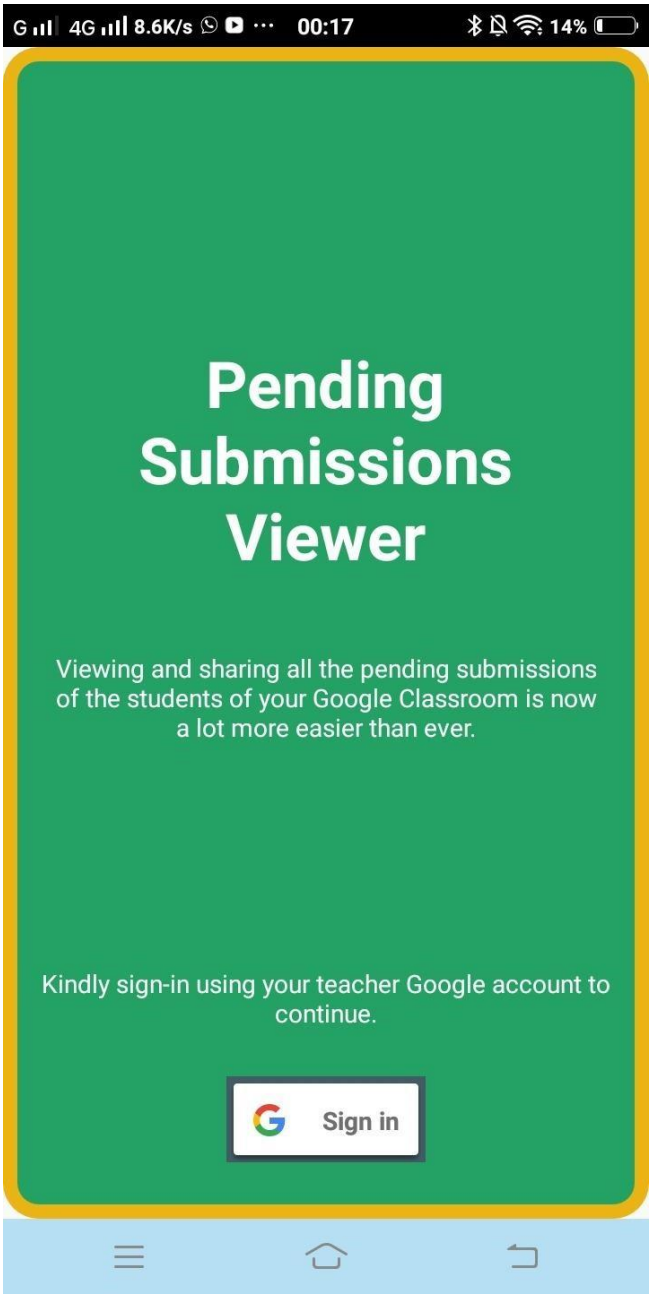


Fig. 6.1.1 B Generating the report

6.2. Pending Submissions Viewer



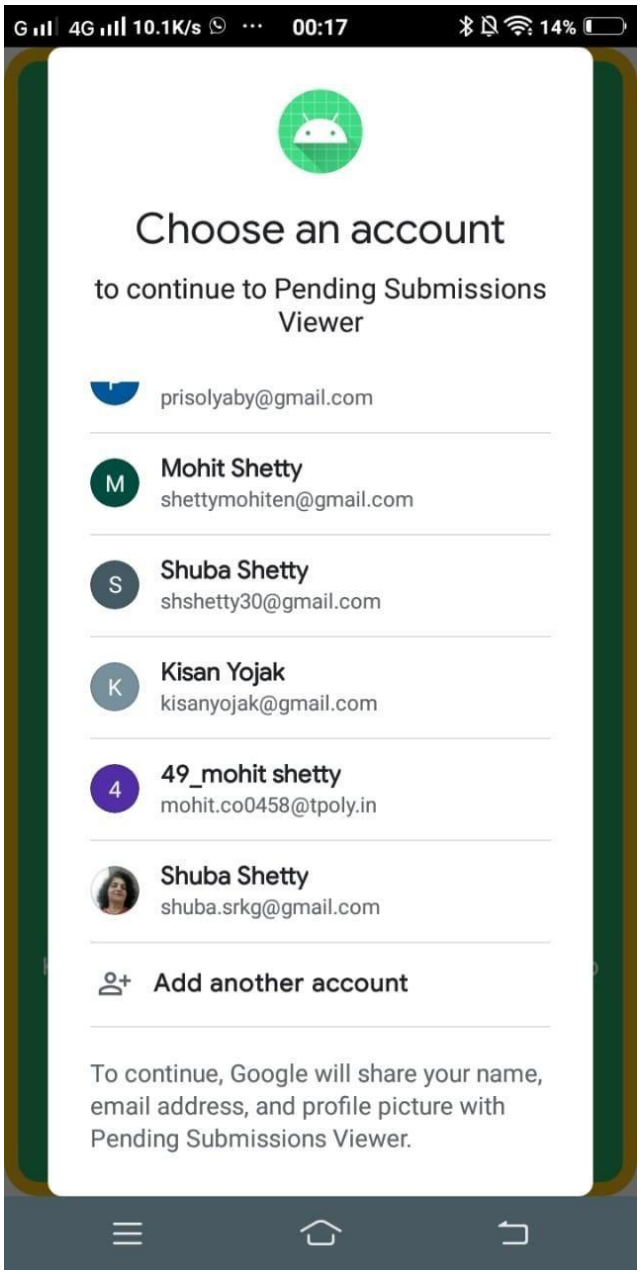


Fig. 6.2.1: Main Activity (Login Screen) **Fig. 6.2.2: Selecting an account for login**

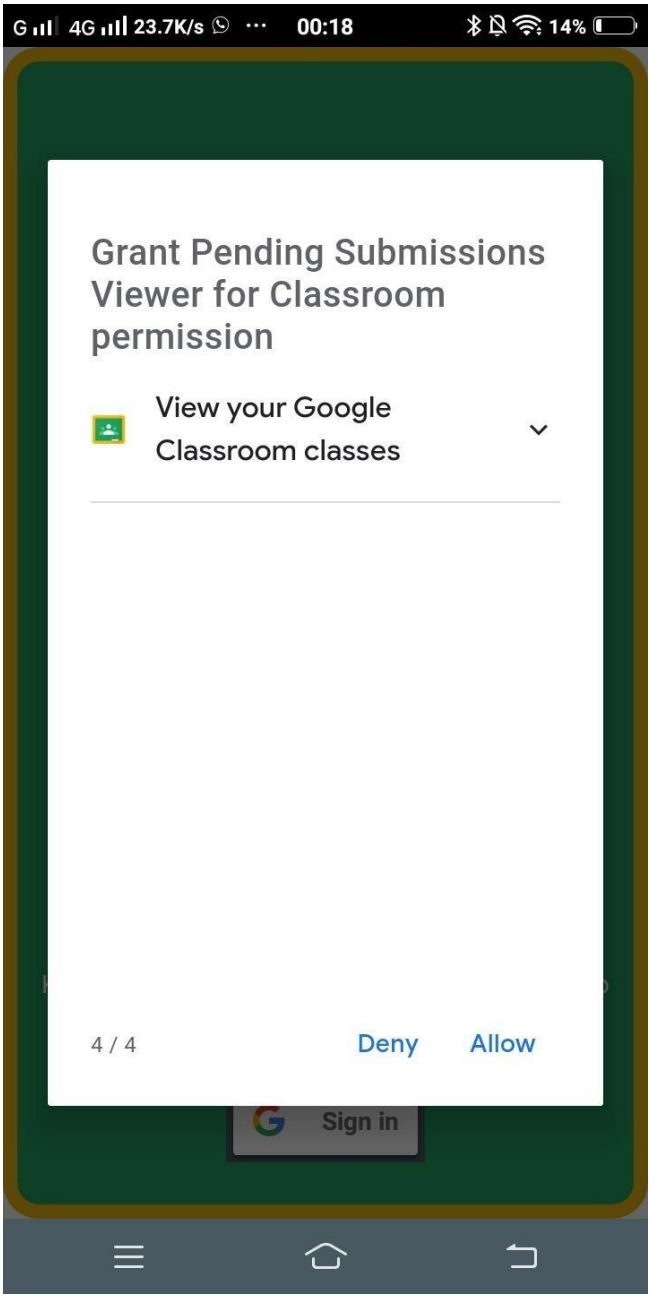


Fig. 6.2.3: Loading Screen (After selecting)

Fig. 4: Accepting permissions for app

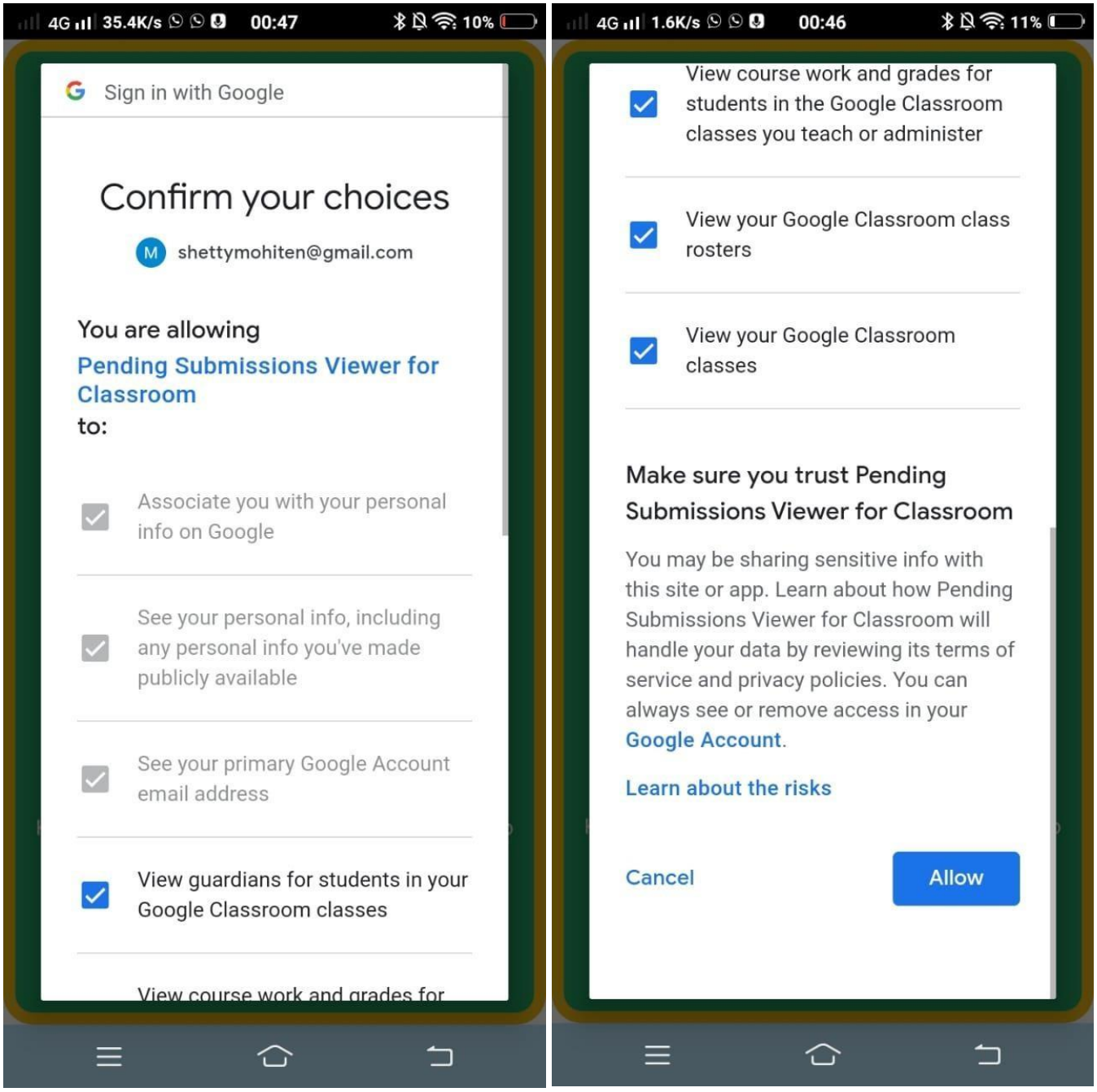


Fig. 6.2.5: Confirming the selected permissions from the user

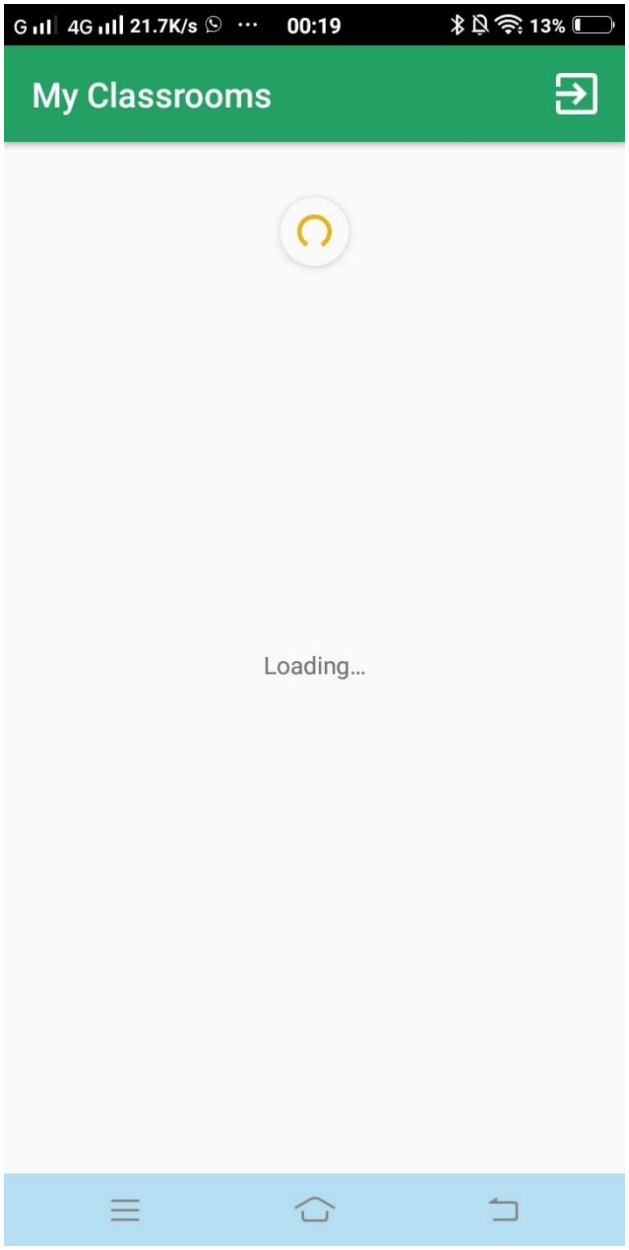


Fig. 6.2.6: Loading all classrooms of user

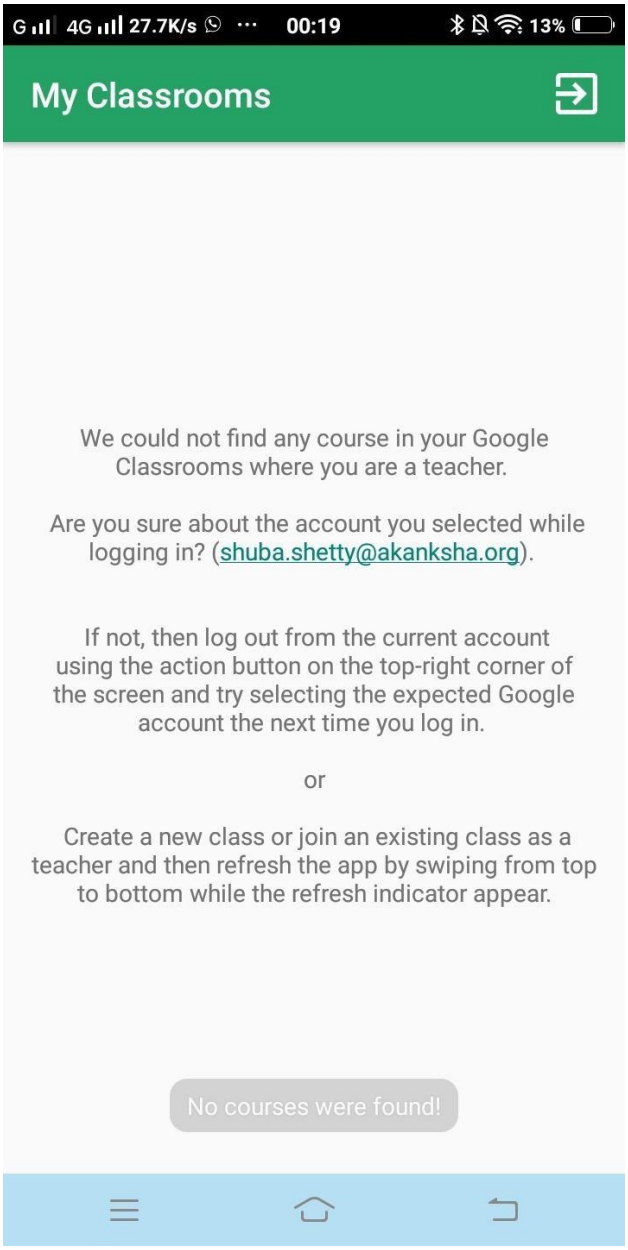
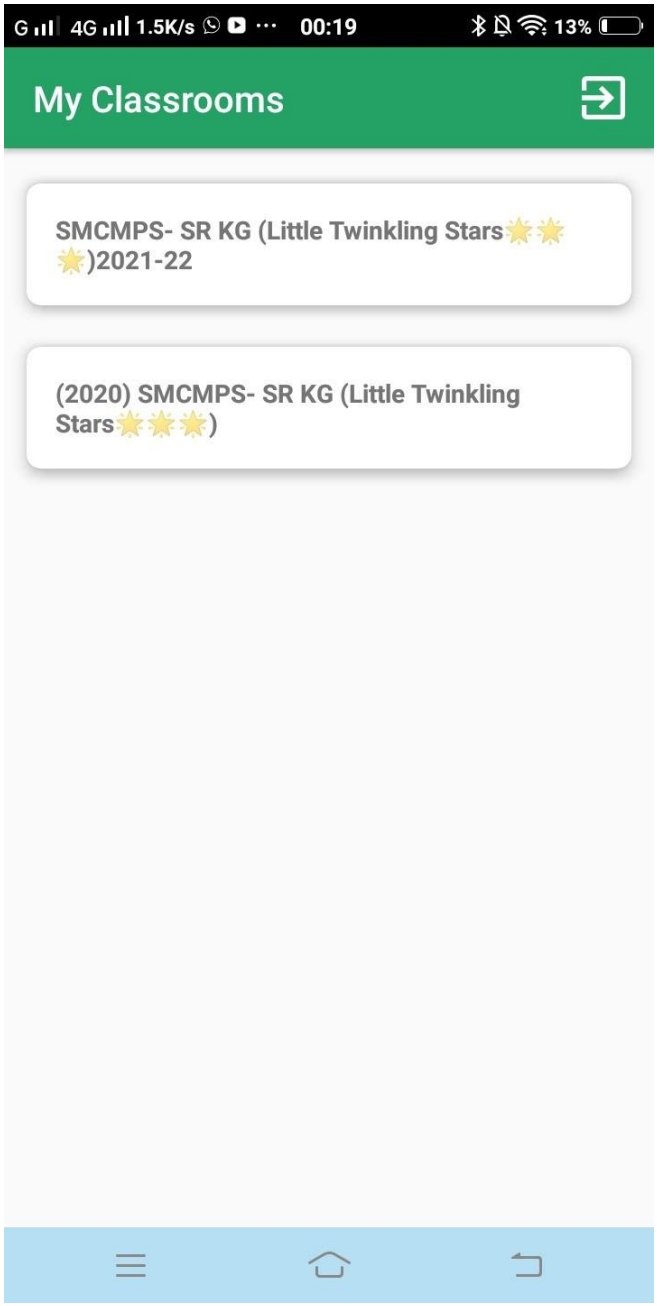


Fig. 6.2.7: No courses/classrooms found



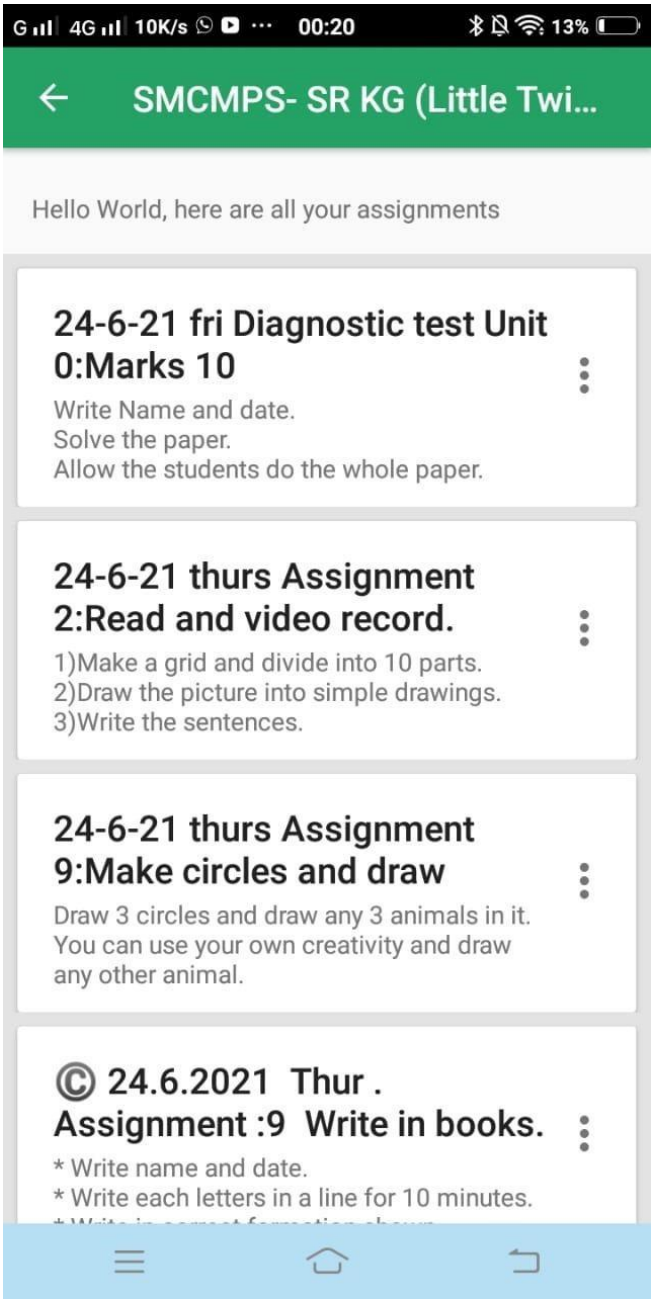


Fig. 6.2.8: Showing all classrooms of teacher **Fig. 9: Listing all assignments**

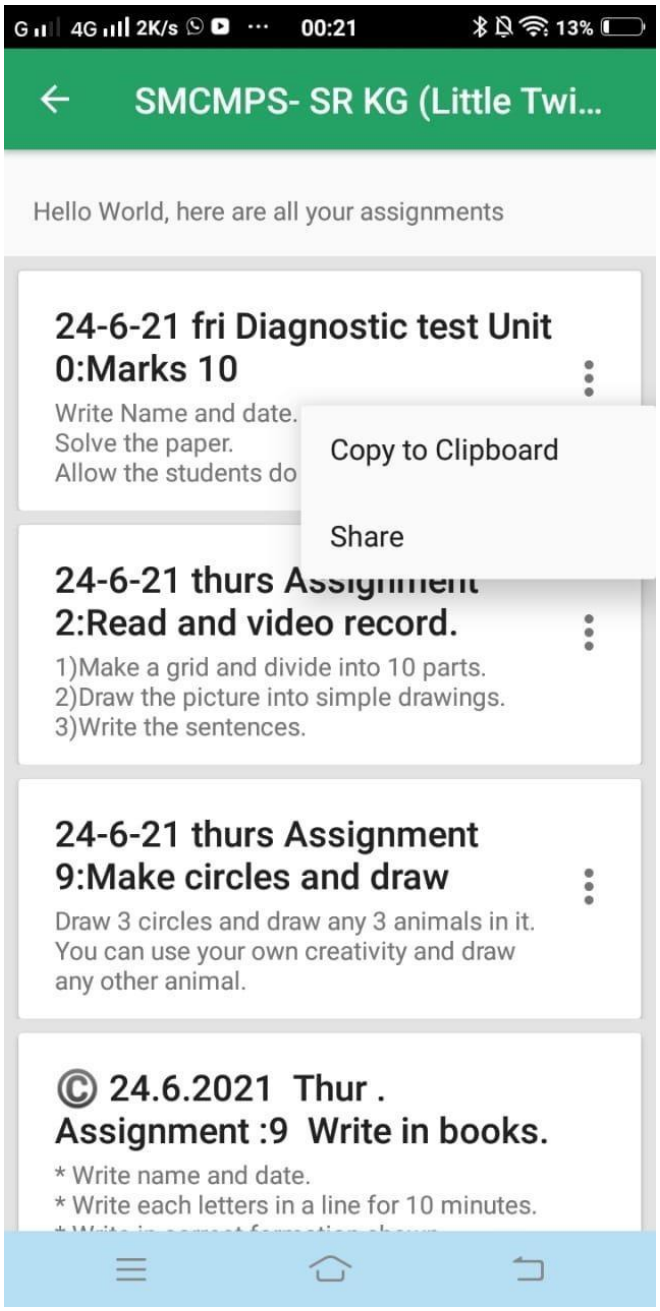


Fig. 6.2.10: Copying/Sharing Options

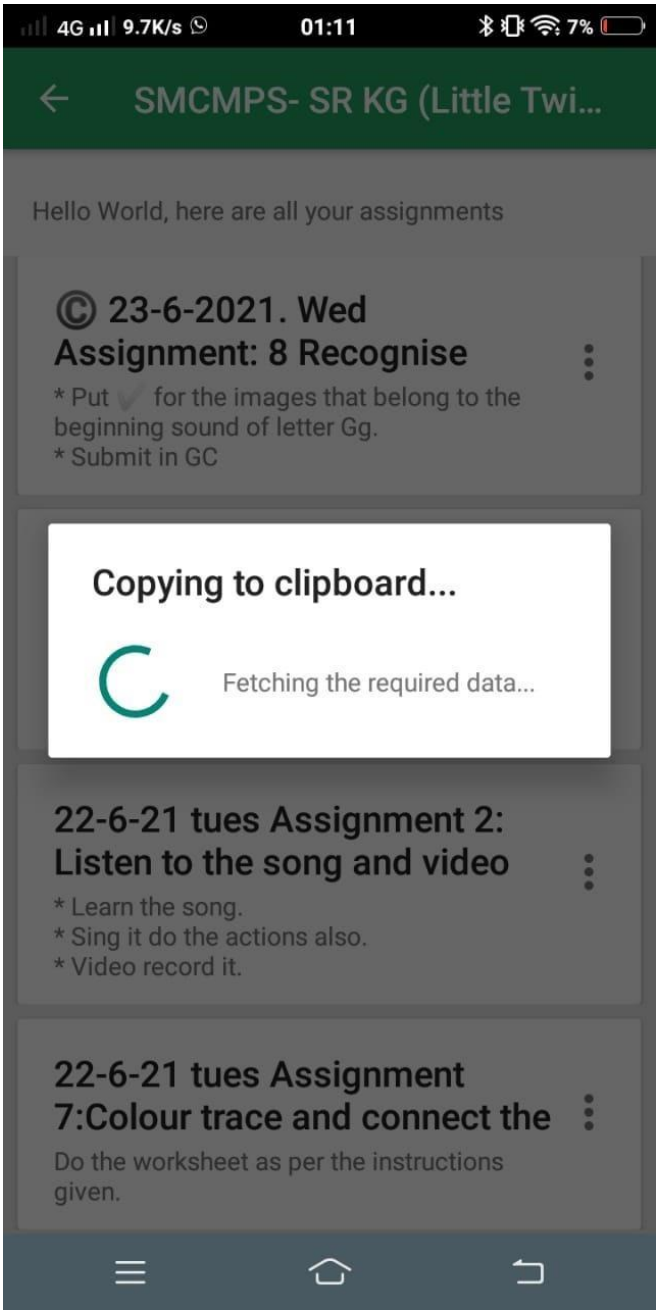


Fig. 6.2.11: Copying to clipboard

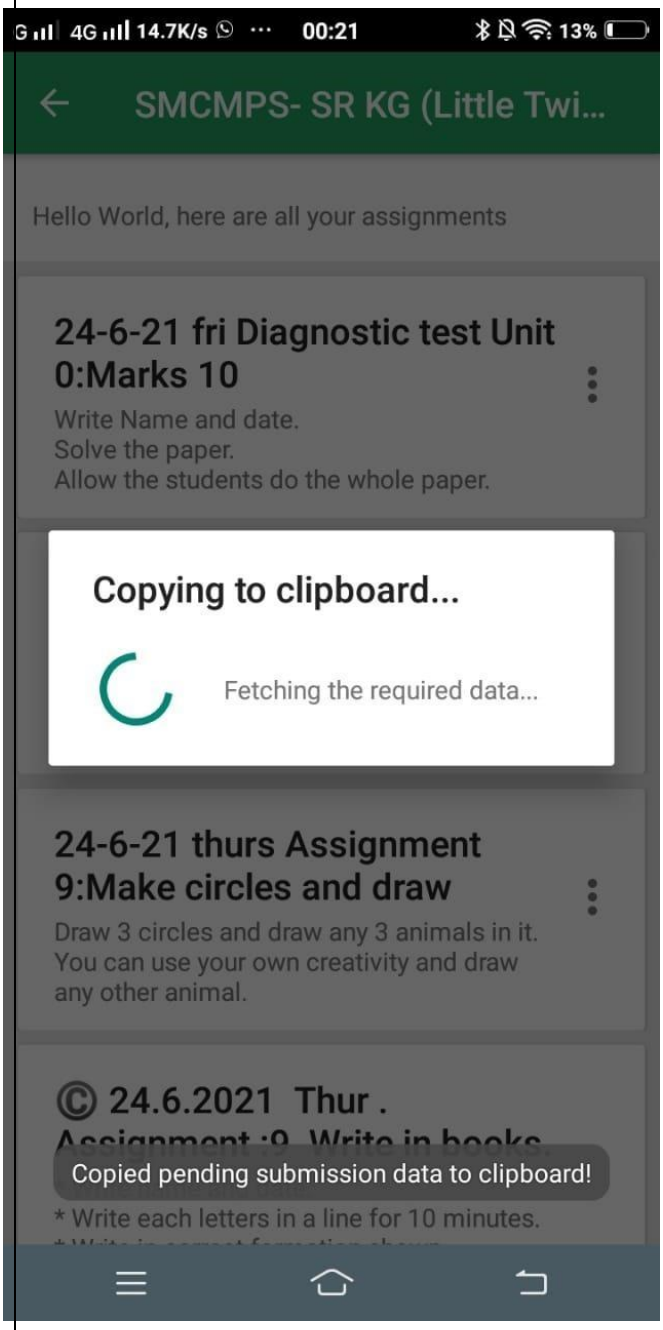


Fig. 6.2.11: Copied to Clipboard data

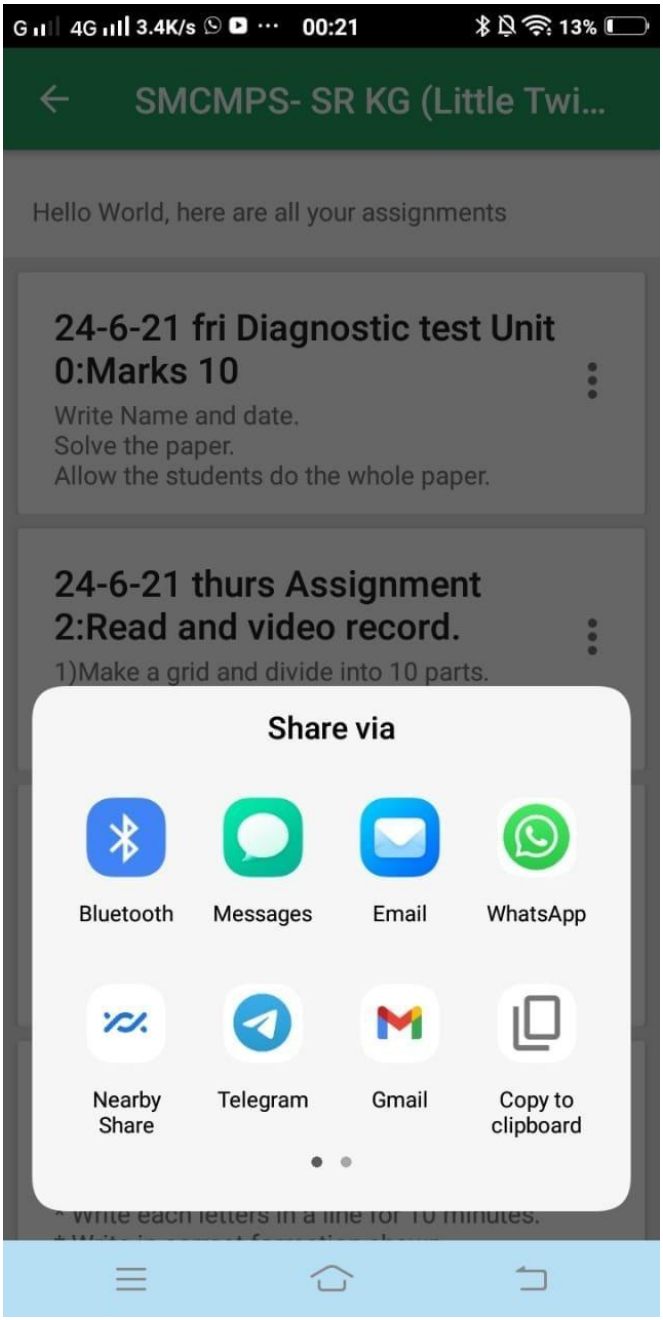


Fig. 6.2.12: Sharing the submission

6.3. MyPHPServer

<u>File Manager</u>

1. Viewing items in a directory

The file manager can be used to view the name, size, last modified date and time, permissions (only for Linux), possible actions that can be performed on the items (files and folders) present in a given directory.

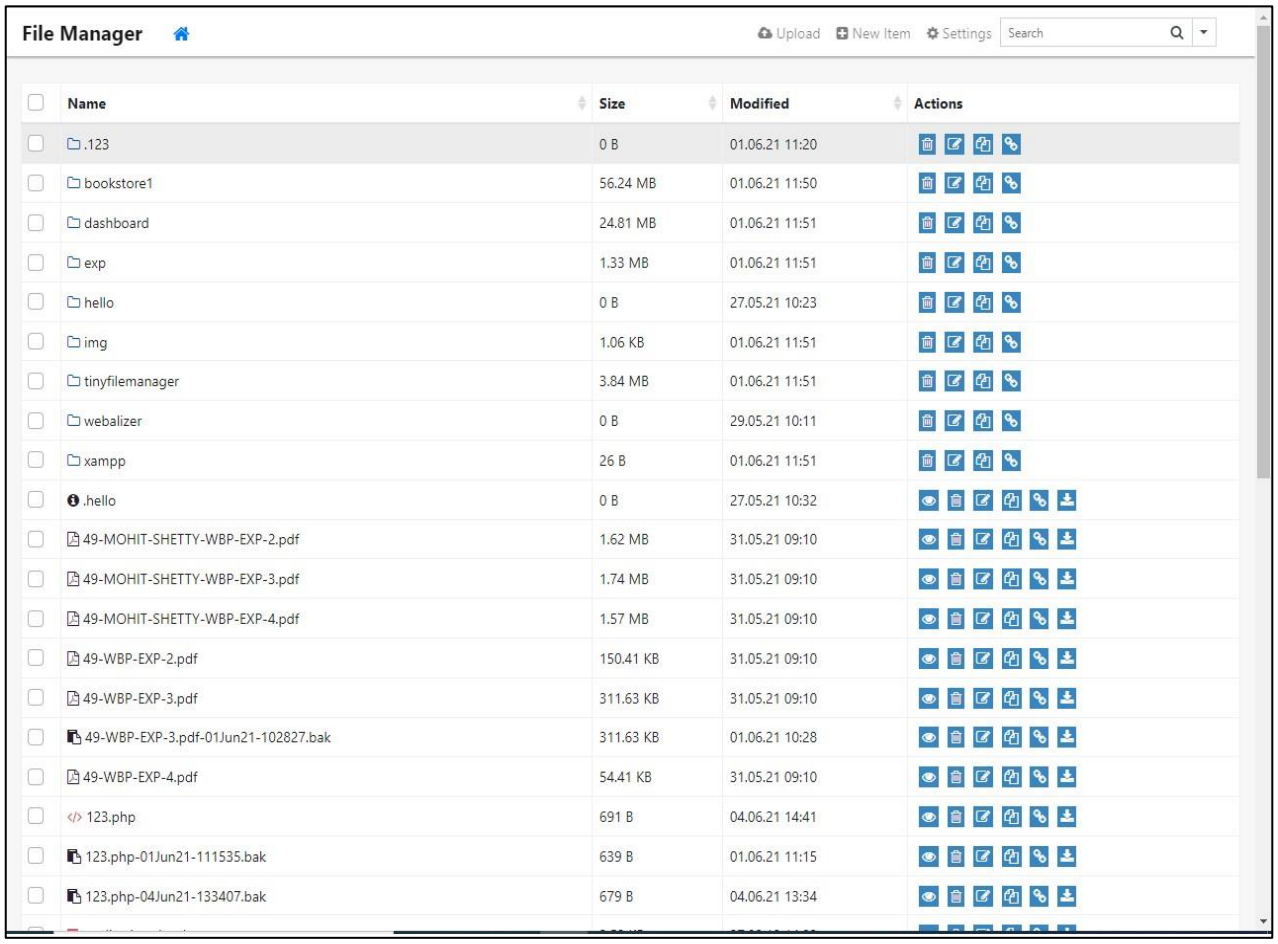


Fig. 6.3.1.1: Displaying the root directory with the help of the File Manager

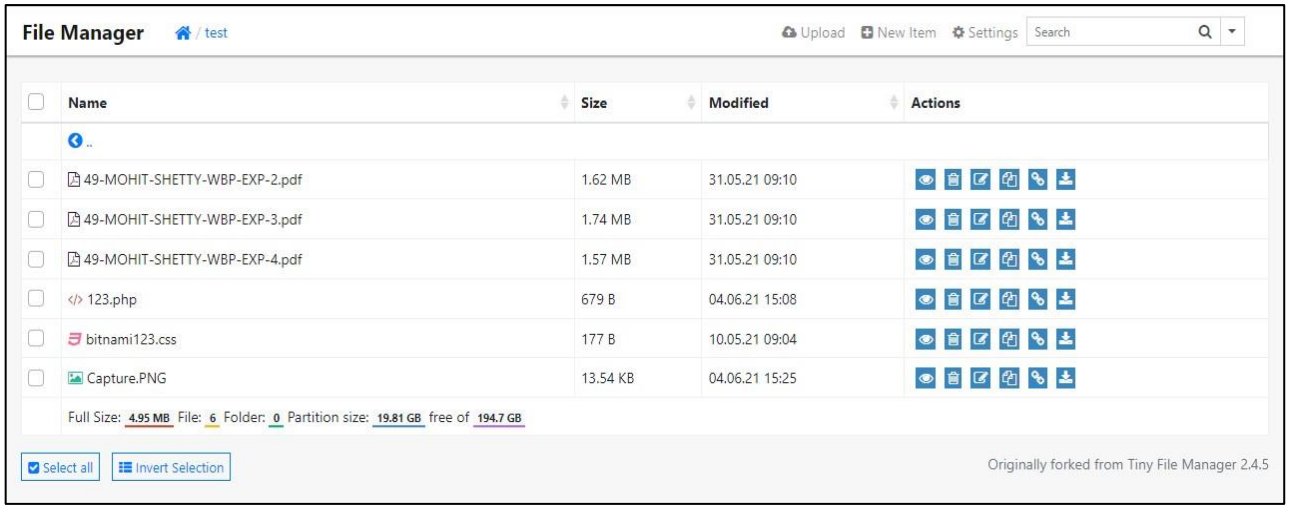


Fig. 6.3.1.2: Displaying a sub-directory with the help of the File Manager

2. Uploading items to the current directory

Since the file system is actually being hosted by a (local) server, we'll need to upload the files via a separate page (that can be viewed by clicking on the *Upload* option present on the header [as visible below]). One can either upload local files by specifying them via drag-and-drop/choosing the files via the default file chooser interface or an external file (uploaded on another server) by specifying its URL to the current directory of the file manager.

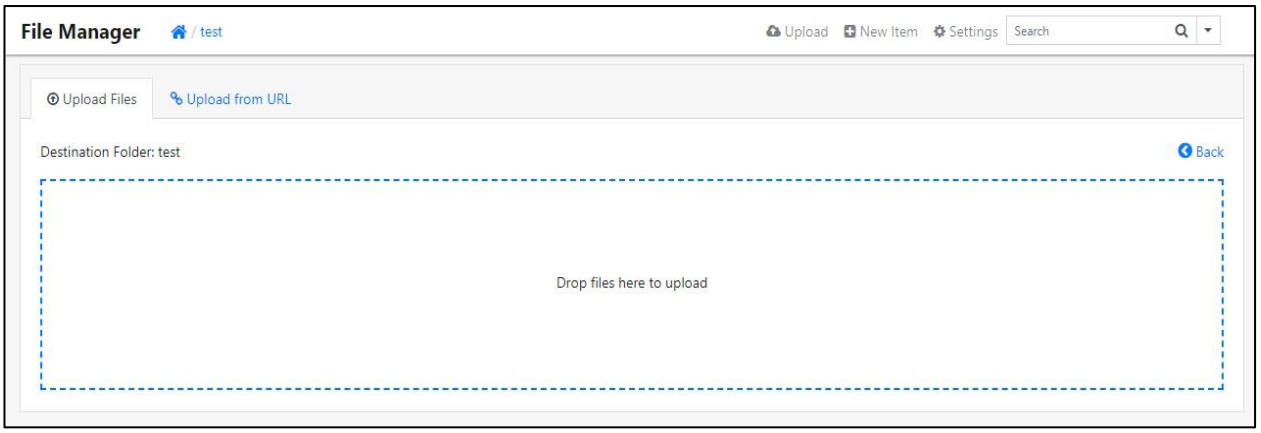


Fig. 6.3.2.1.1: Section to upload local files to the current directory of the file manager

Fig. 6.3.2.1.2: Uploading 3 files to the current directory ('test') of the file manager

Fig. 6.3.2.2.1: Section to upload an external file via its URL to the current directory of the file manager

Fig. 6.3.2.2.2: A success message displaying that the file was successfully uploaded

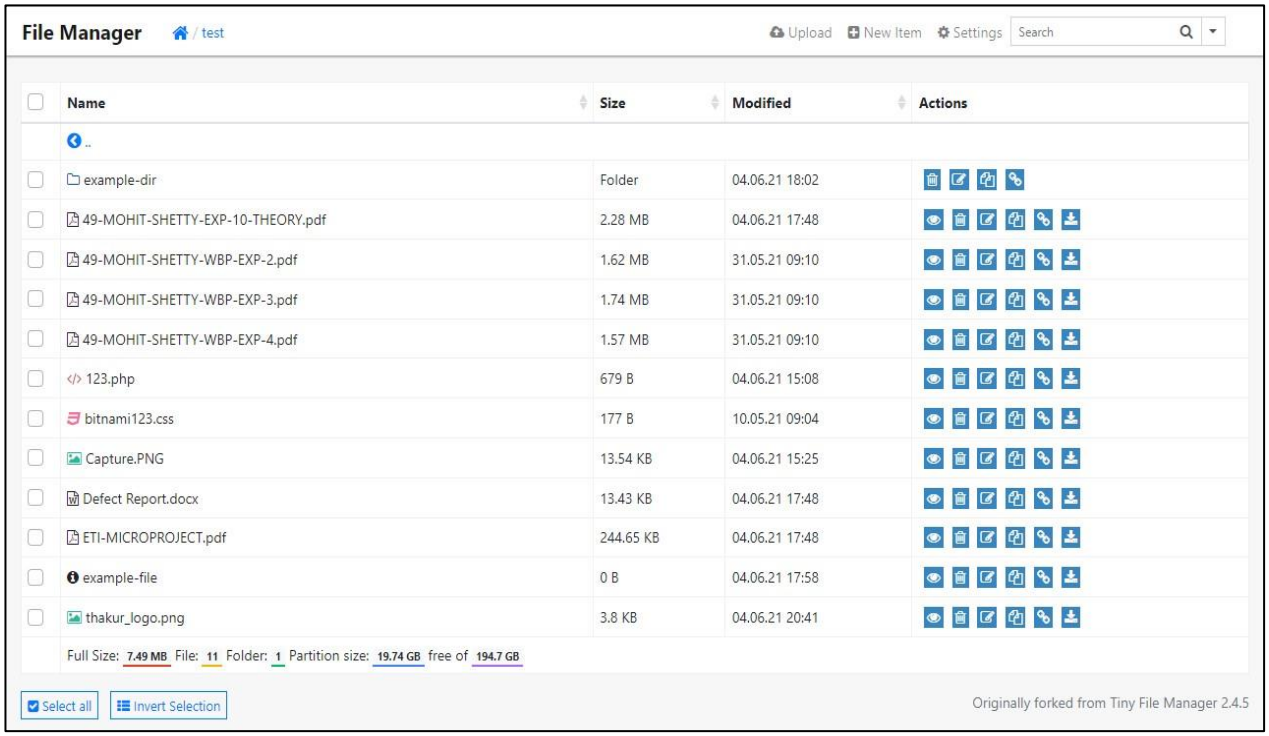


Fig. 6.3.2.3: Displaying the uploaded files in the 'test' directory of the server

3. Creating a file and folder

A file and folder can easily be created by heading over to the ‘New Item’ option, which would open a popup. In the popup, choose whether you want to create a file or folder, typing in its name and then clicking on the ‘Create Now’ option at the bottom of the popup.

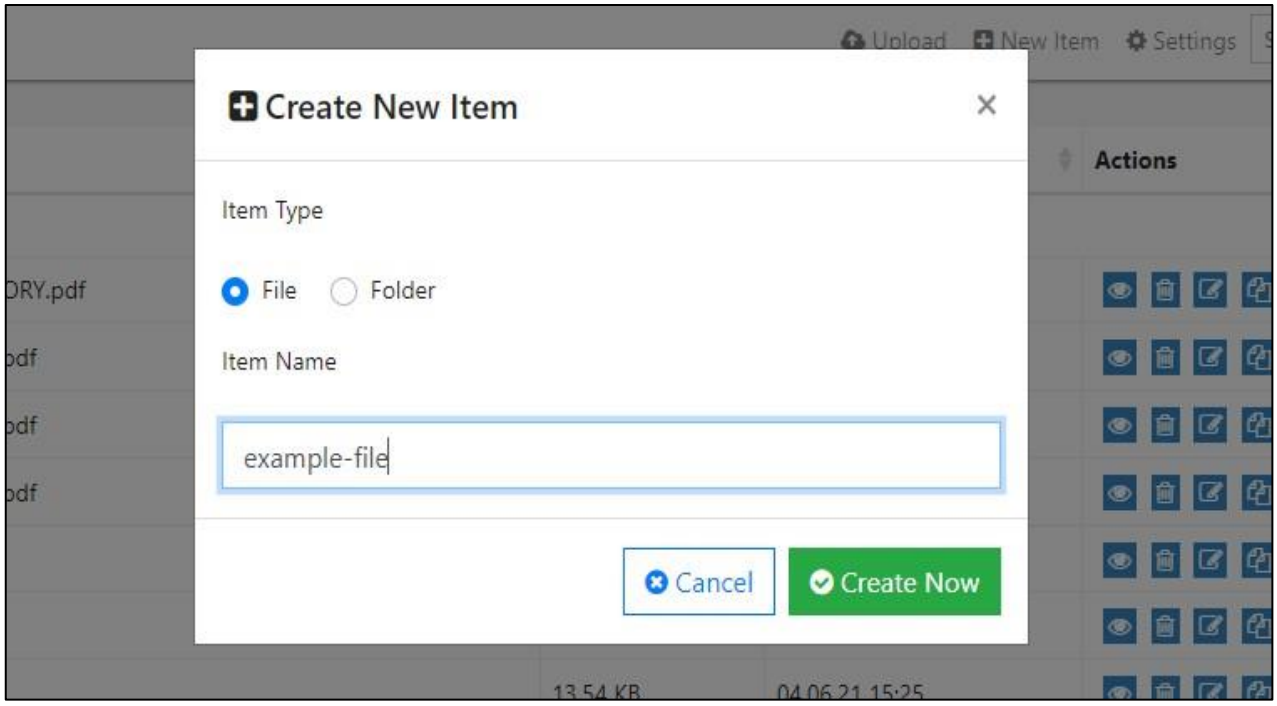


Fig 6.3.3.1.1: Creating a new file from the ‘New Item’ option

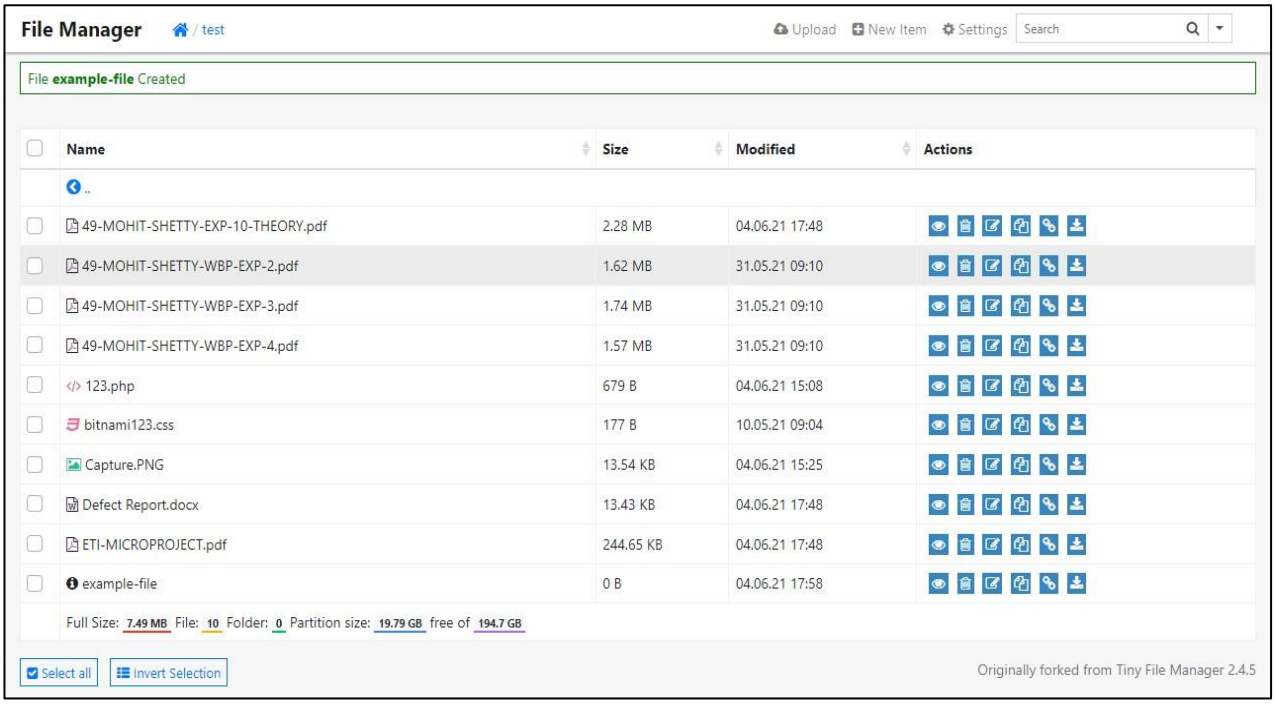


Fig. 6.3.3.1.2: Success message after adding the file ‘example-file’ to the ‘test’ directory

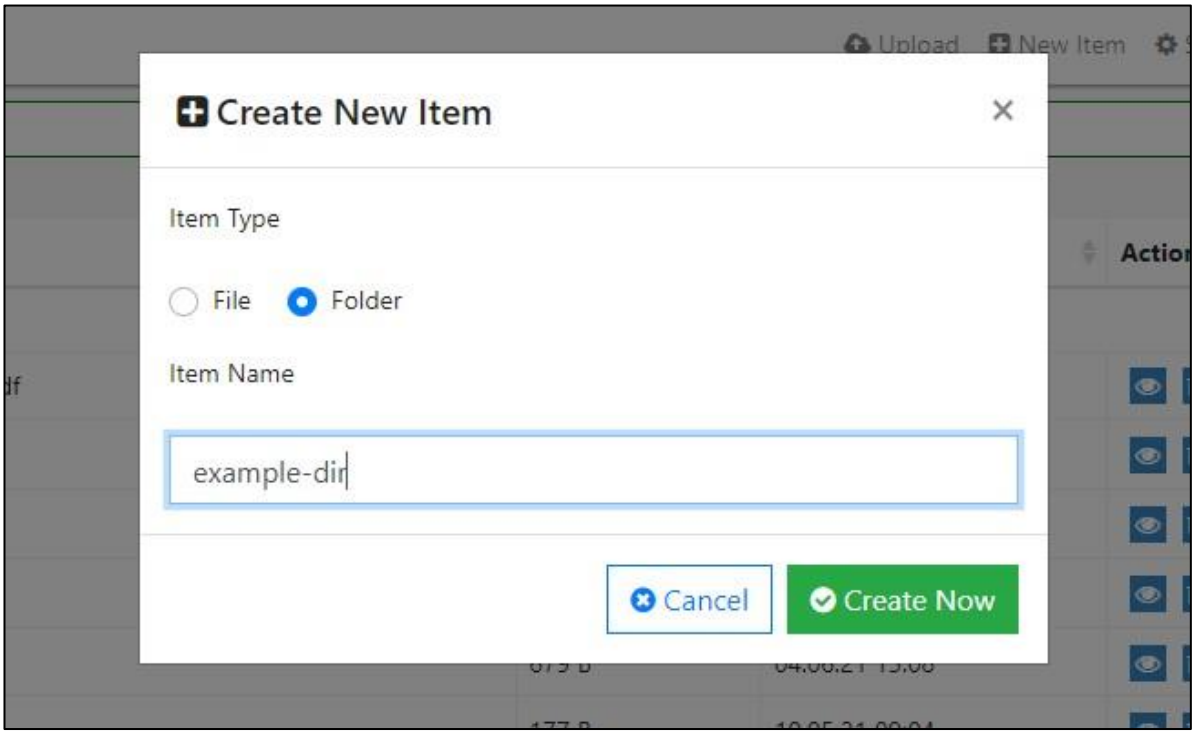


Fig. 3.2.1: Creating a new folder 'example-dir' from the 'New Item' option

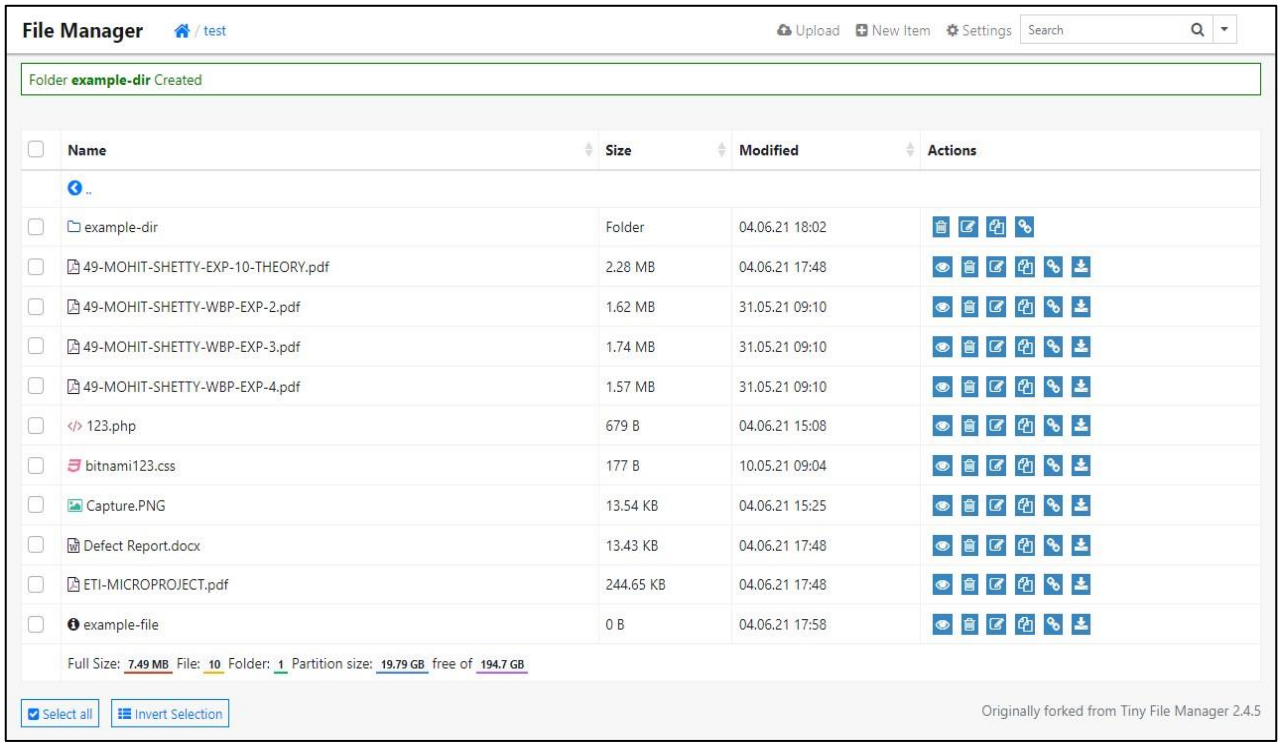


Fig. 6.3.3.2.2: Success message after adding the dir. 'example-dir' to the 'test' directory

4. Modifying the settings of the file manager

The file manager has a settings section as well where one could,

1. Change the default language of the file manager from a list of supported language
2. Enable/Disable error reporting for the UI (of the file manager) being rendered.
3. Tell whether the file manager whether or not hidden items should be displayed (basically files/folders starting with a dot [.])
4. (Only for Linux) Tell the file manager to (not) display the permissions for a given item.
5. Decide whether the size of a folder should be calculated (for the 'Size' column) or not.

Note: The changes won't reflect until the 'Save' option is clicked

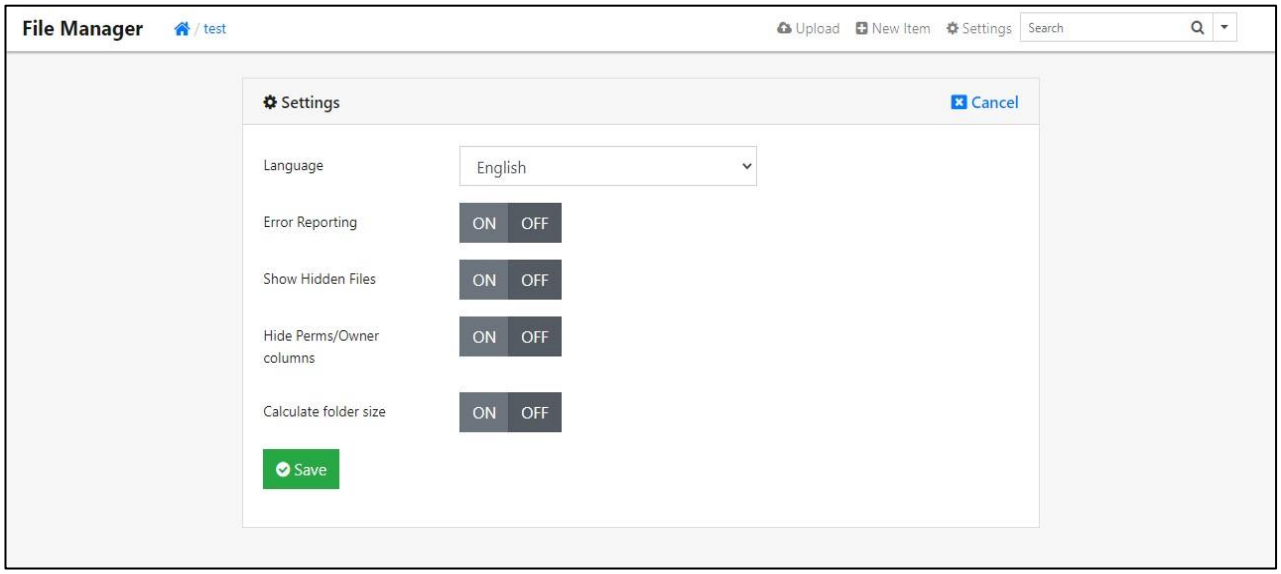


Fig. 6.3.4: Settings page of the application

Note: All the options given above are **OFF**.

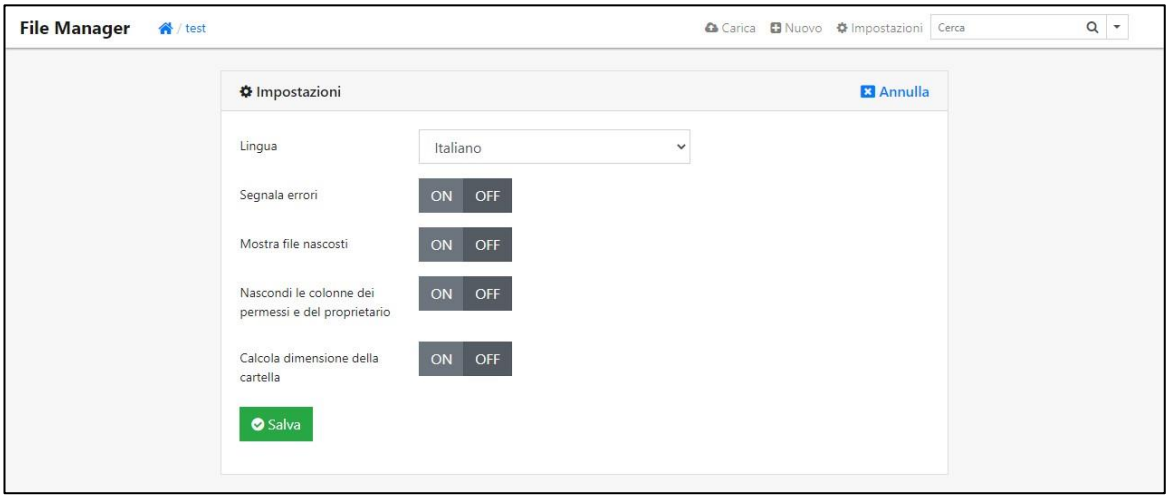


Fig. 6.3.4.1: Changing the default language of the app (to Italiano)

The application has been thoroughly tested and now redirects to the root directory in case of an unexpected error. So error reporting has only been only added for testing purposes (while adding new features [in the future]).

To test the hidden items feature, a hidden file named **.some-hidden-file** was created.

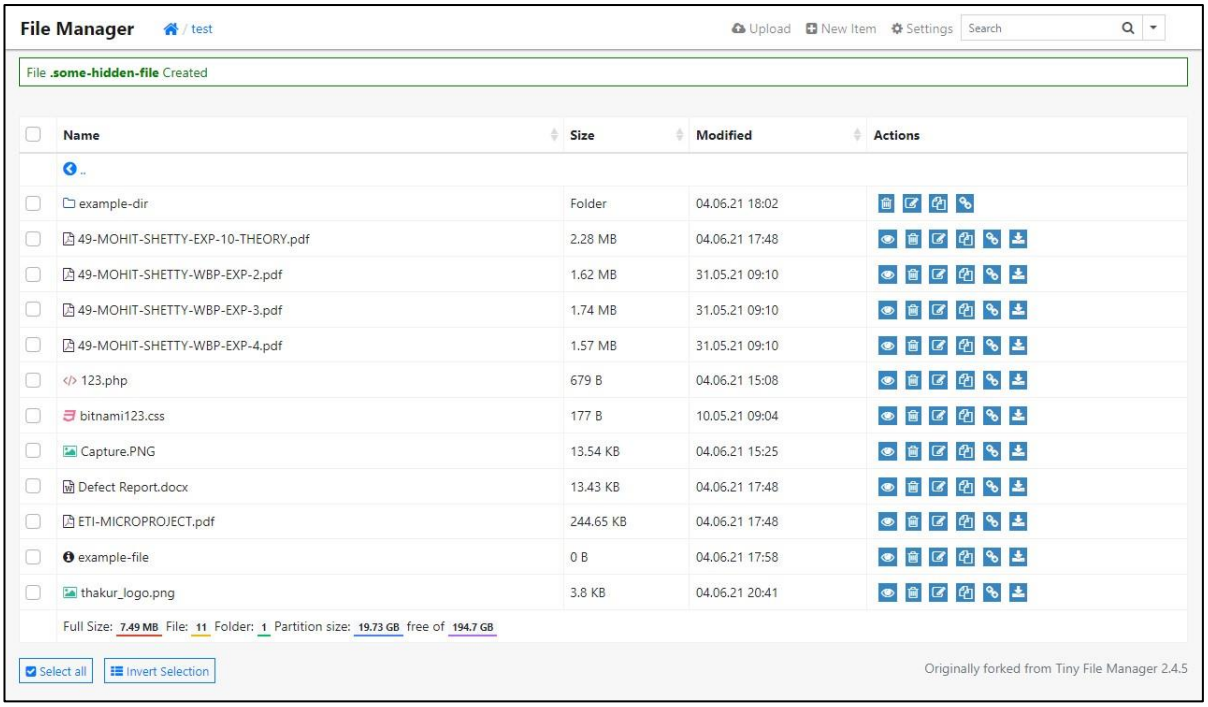


Fig. 6.3.4.3.1: Creating the file when the 'Show Hidden Files' option was disabled (by default)

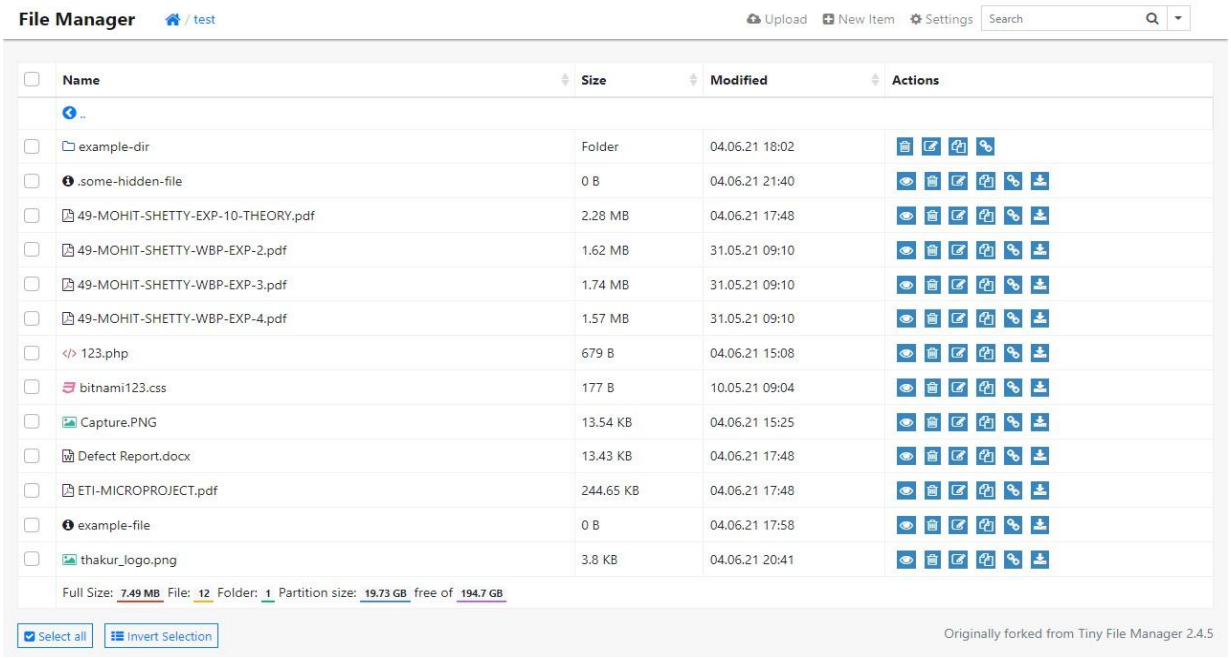


Fig. 6.3.4.3.2: The file is now visible, after enabling the 'Show Hidden Files' option in the settings

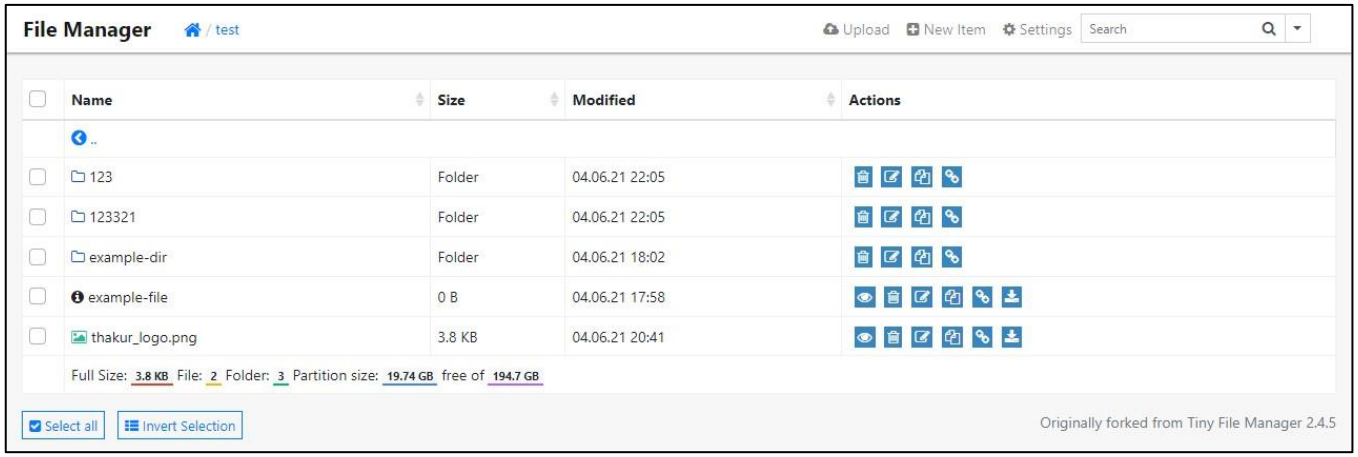


Fig. 6.3.4.5.1: Displaying all files when the 'Calculate folder size' is off/disabled

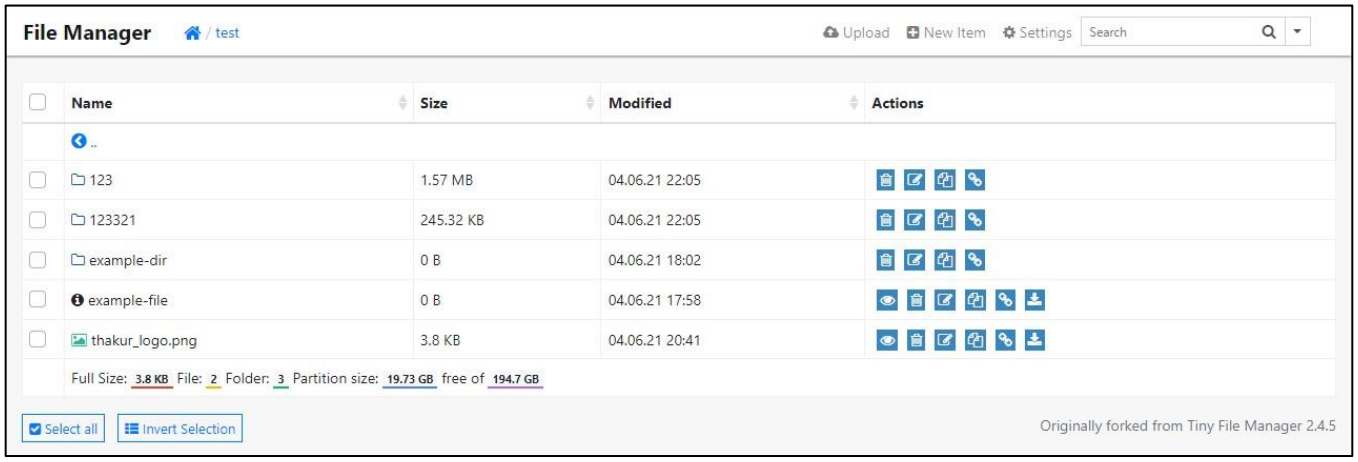


Fig. 4.5.2: Displaying all files when the 'Calculate folder size' is on/enabled

5. Searching for an item

An item can easily be searched for by partially typing its name onto the search bar. The relevant files would then be displayed as the main contents of the page.

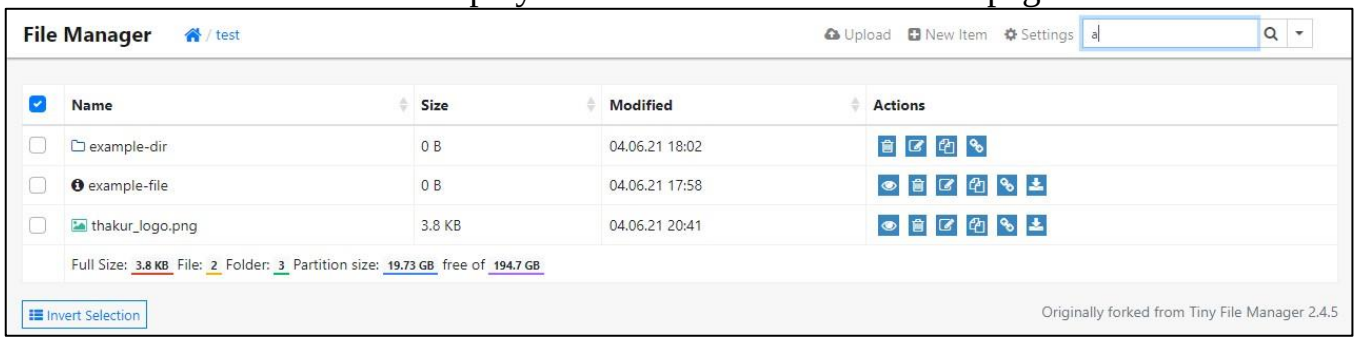


Fig. 6.3.5.1: Displaying all the files in the current directory (containing ‘a’)

Fig. 6.3.5.2: A situation when no records are found

6. Previewing a file

The file manager also provides an option to preview a file. Generally, the file is either rendered as a img or in an iframe (e.g. PDF) or if it is an image file or in the form of highlighted texts, if the file is a recognized text/code file or at least as some plain text.

Fig. 6.3.6.1: Previewing a code/script file using the Preview option (eye icon)

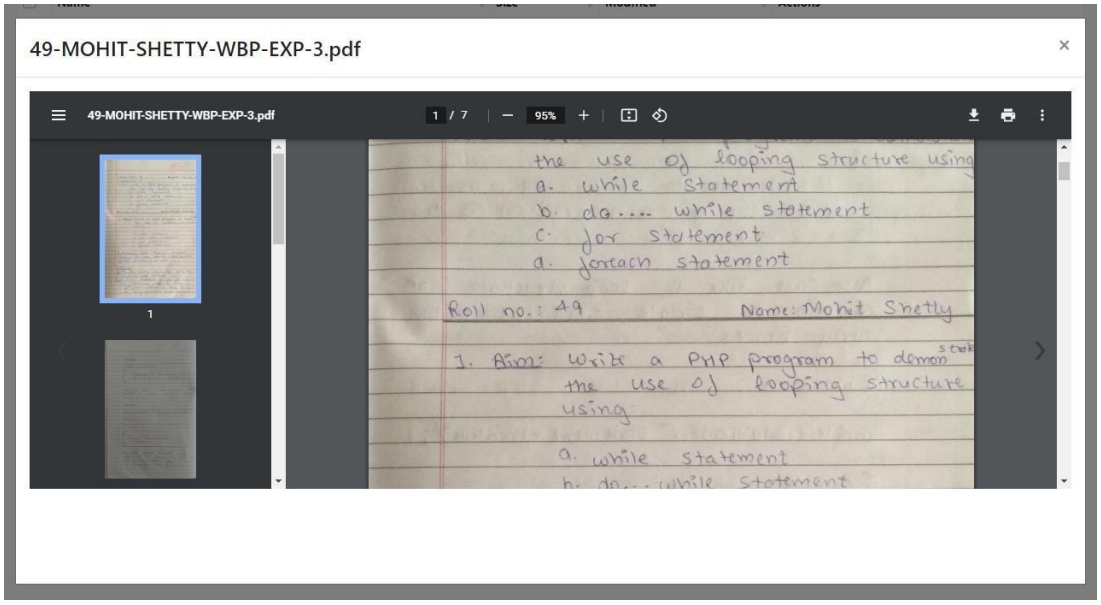
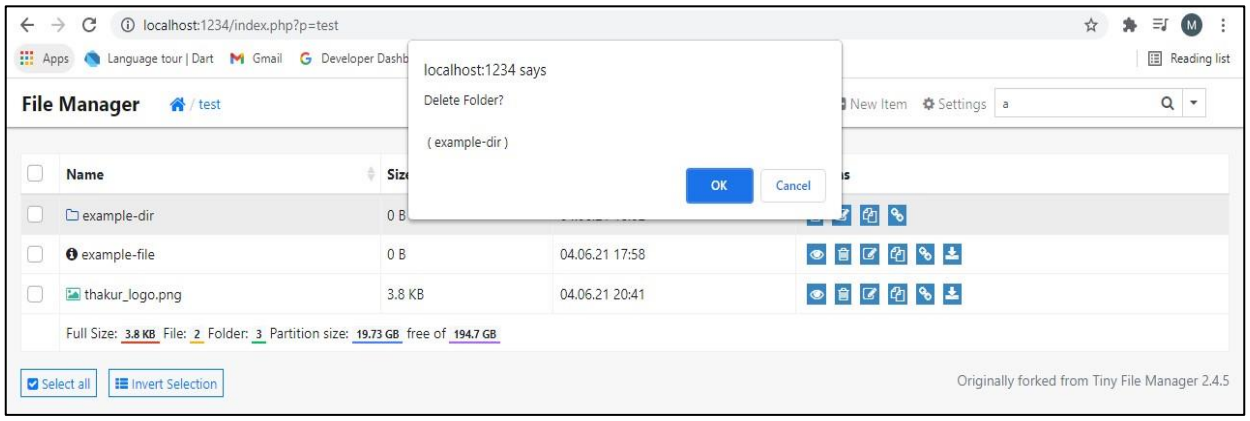


Fig. 6.3.6.2: Previewing an PDF file

7. Deleting an item

An item being displayed by the file manager can be easily deleted by clicking on the ‘Delete’ option under the ‘Actions’ column. You will be prompted with a confirm dialog, before the file actually gets deleted. If you choose to proceed, the file shall get deleted.



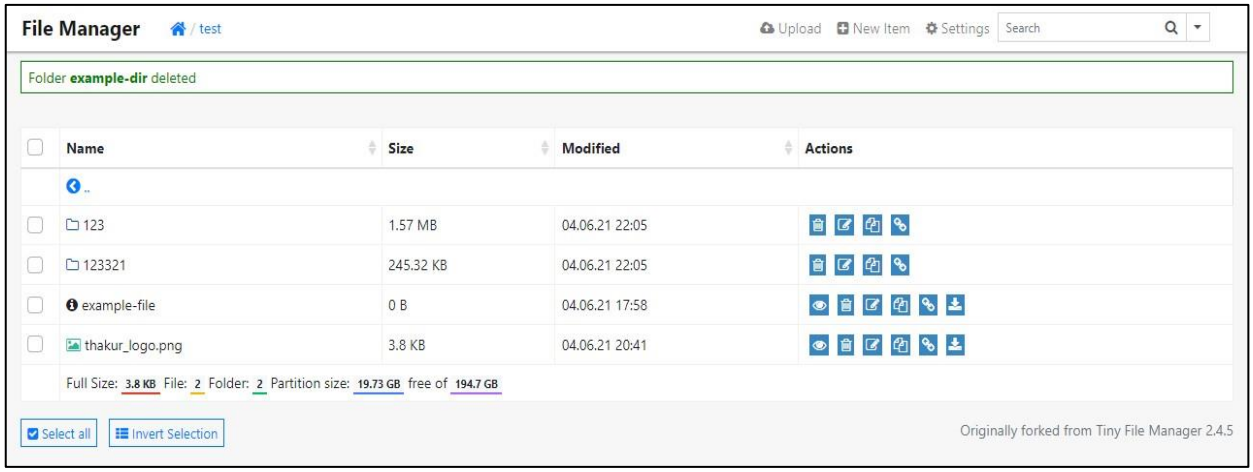


Fig. 6.3.7.1, 6.3.7.2: Deleting a directory called 'example-dir' present in the 'test' directory

8. Renaming an item

An item can be renamed by two ways:

1. By quickly double tapping on the name of the item, editing the filename in the text field to the new one and then pressing enter or losing focus of the field. 2. By clicking the 'Rename' button in the action section, waiting for a dialog to appear, entering the name of the new file in it and confirming it.

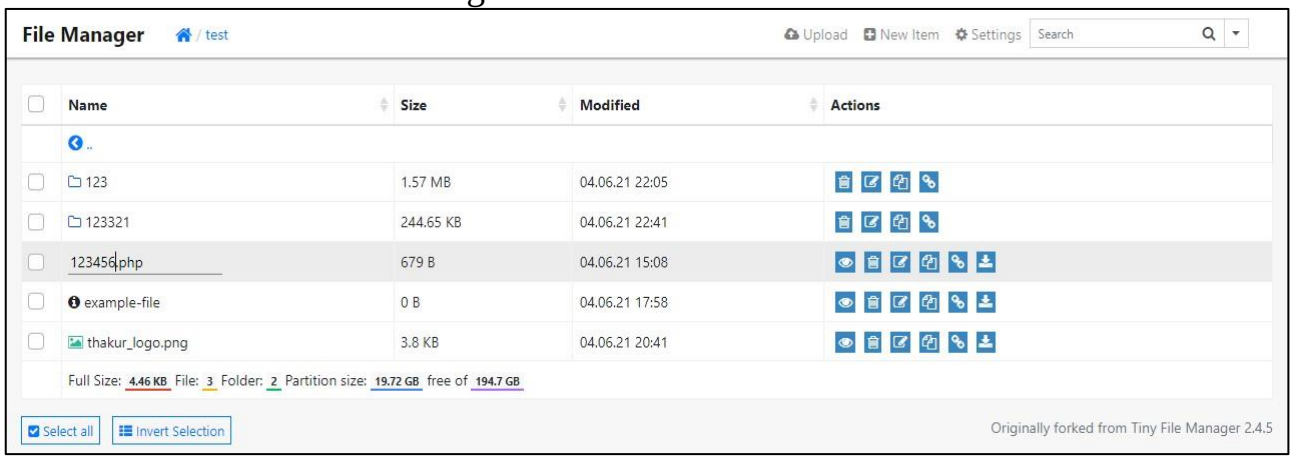


Fig. 6.3.8.1: Renaming an existing file with the first method (double tap/click)

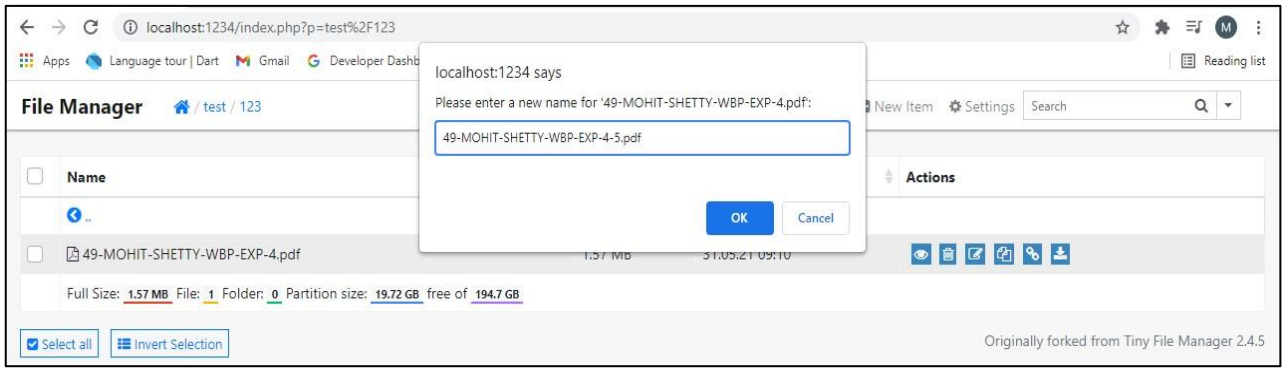


Fig. 6.38.2: Renaming an existing file with the second method (Action column)

9. Moving/Copying an item from one directory to another

One can easily copy an item from one directory to another by clicking on the ‘Copy’ button (actions). Once done so, a menu shall appear where you shall be requested to navigate to the destination folder. Once you are done navigating, choose whether you want to move/copy the item and the test shall be taken care of by the file manager.

Fig. 6.3.9: Choosing the destination path of the item to be moved/copied

10. Heading over to the direct link of a file

One can easily navigate to the direct link of a file by clicking on the ‘Direct Link’ option present under the ‘Actions’ column.

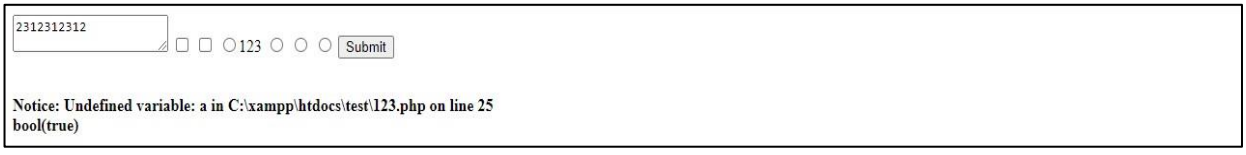


Fig. 6.3.10: The page that opens after choosing the Direct Link option for 123.php

11. Downloading a file

By clicking on the ‘Download’ option/icon under the ‘Actions’ column, one can easily download a file from the server to make it locally accessible.

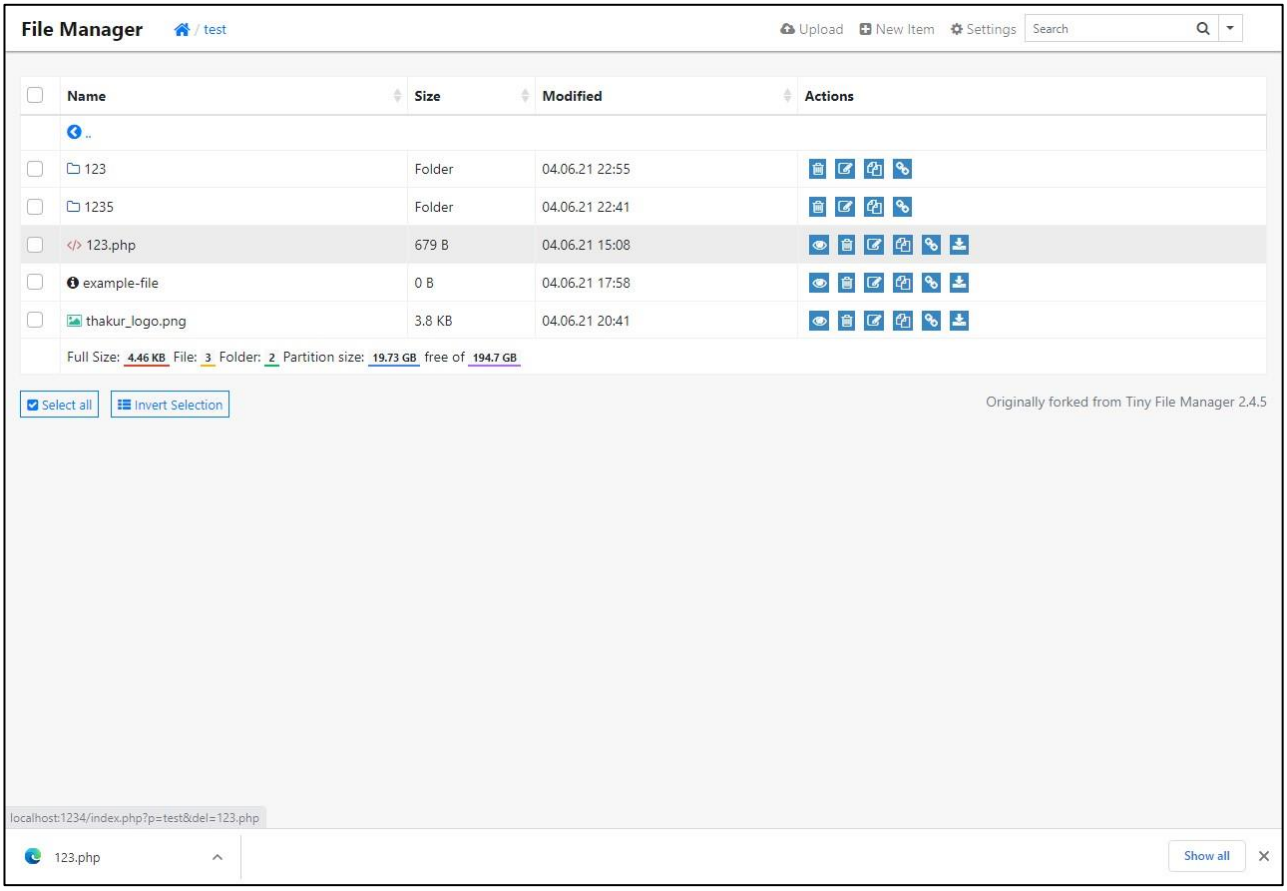
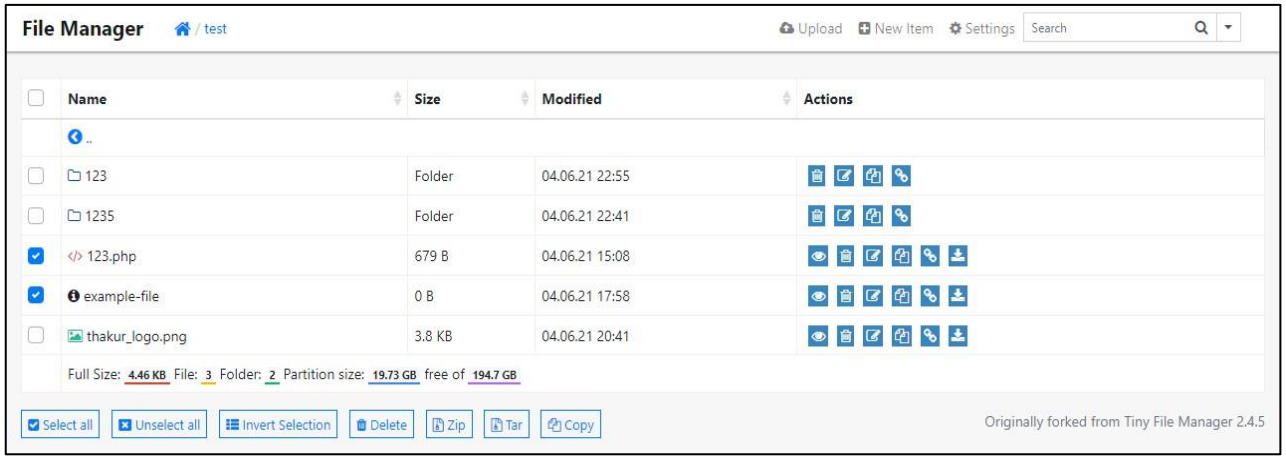


Fig. 6.3.11: Downloading a PHP script with the help of the file manager

12. Selecting multiple items

One can easily select multiple items by clicking on the checkbox at the left hand side of the items to be selected.



12. Inverting the selection

One can easily invert the selection by clicking on the '**Invert Selection**' option.

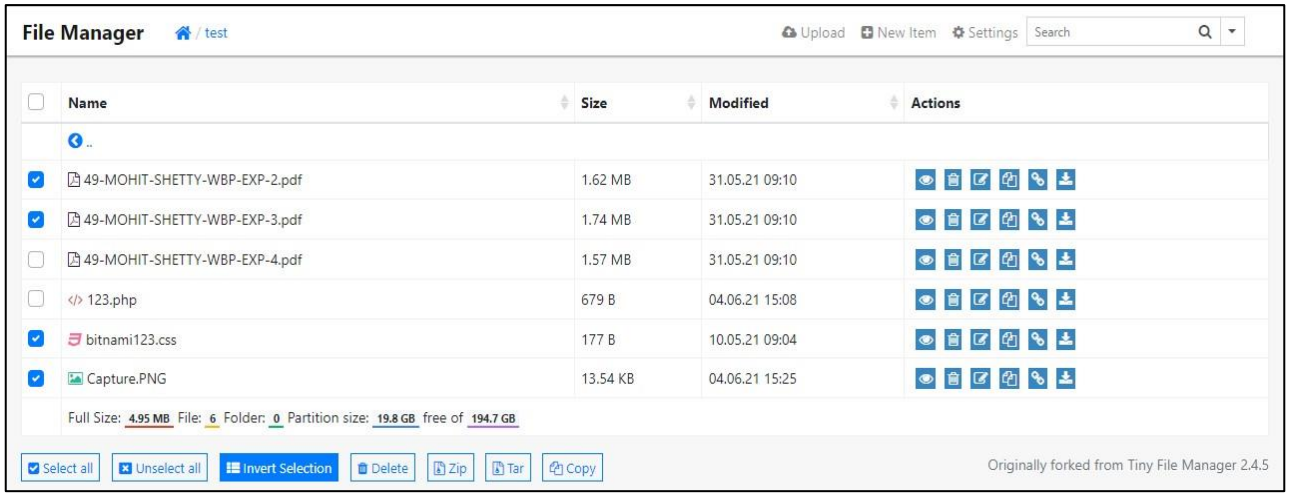


Fig. 6.3.12: Inverting all the selected items using Invert Selection option

13. Selecting all the items

One can easily select all the items by clicking on the **'Select All'** option or checking the checkbox present at the header of the table.

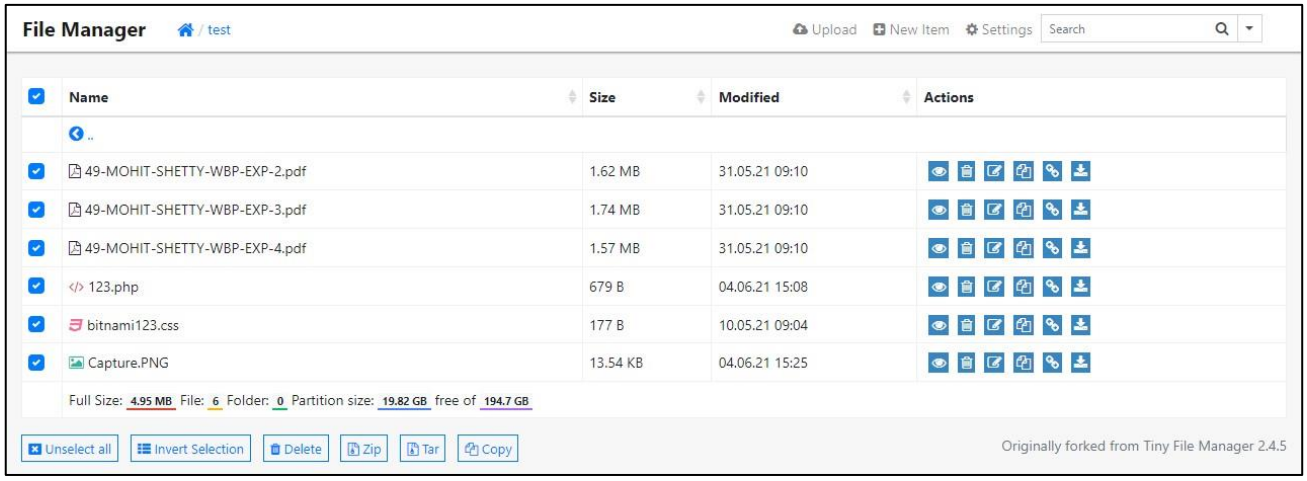


Fig. 6.3.13: Selecting all items in a directory

14. Deleting multiple items

One can delete multiple items by selecting the items to be deleted and then clicking on the 'Delete' option present in the footer when done (alternatively, the keyboard button 'Delete'/'Del' can be pressed as well). When a confirmation dialog appears, the user can click on 'OK' to proceed.

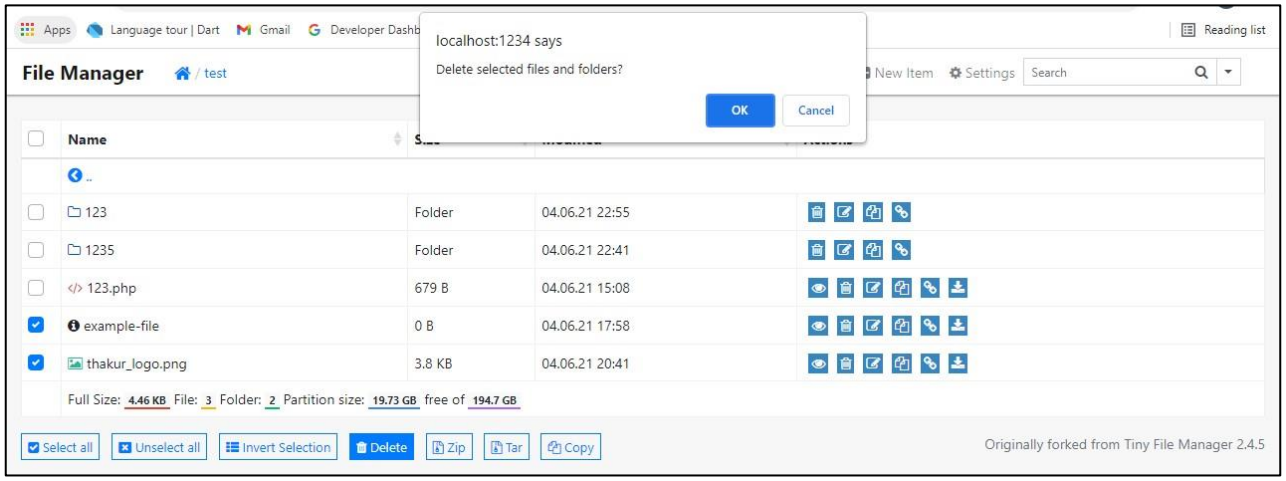


Fig. 6.3.14: A confirmation dialog ensuring that the user had not accidentally pressed the 'Delete' button

15. Compressing multiple items (using zip/tar)

One can compress multiple items into a single file by selecting the items to be compressed and then by choosing the target extension when done. When a confirmation dialog appears asking 'Create Archive?', click on 'OK' to proceed.

Note: The process might take some time considering the number of files and their sizes

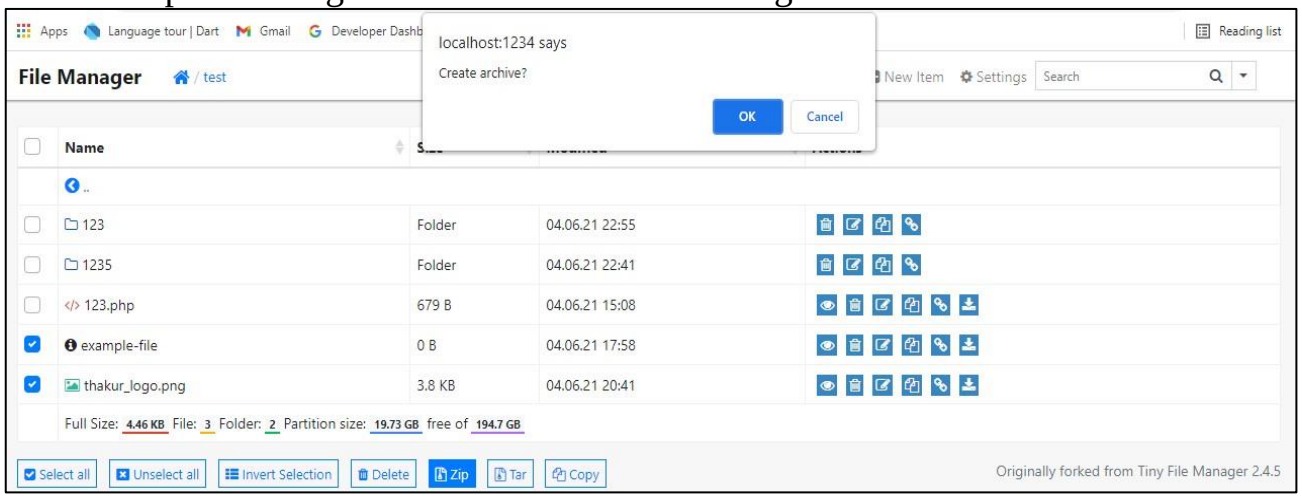


Fig. 6.3.15: A confirmation dialog confirming that the user had clicked the Zip button

16. Copying/Moving multiple items

One can copy/move multiple items by selecting the items to be copied/moved and then clicking on the ‘Copy’ option present in the footer when done. A menu shall appear as the main contents of the page (as shown below). Type the path to the file and check the ‘Move’ checkbox if the user intends to move the file and click ‘Copy’, when done.

Fig. 6.3.16: The menu that appears after clicking on the ‘Copy’ button for multiple files

15. Viewing an item

An item can easily be viewed by clicking on it. If it is a directory, then the file manager would display the contents of that path, else if it is file the meta data and the logical contents of the file.

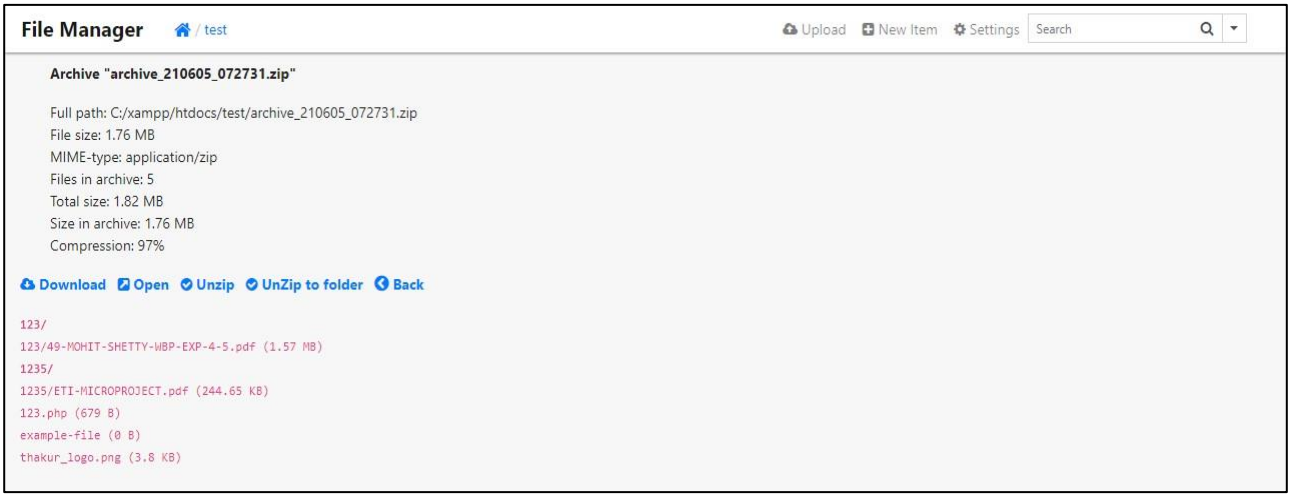


Fig. 6.3.15.1: Viewing the contents of a zip file along with its metadata

Fig. 15.2: Viewing an image file

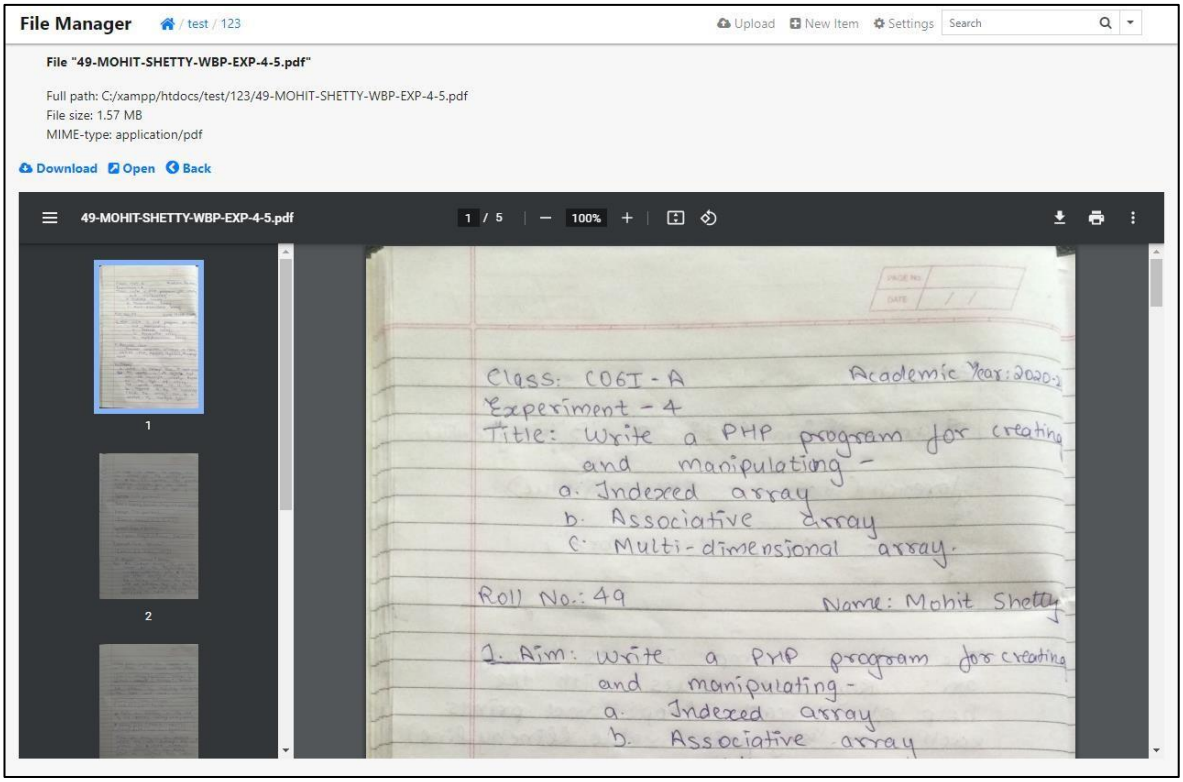


Fig. 6.3.15.3: Viewing a PDF file

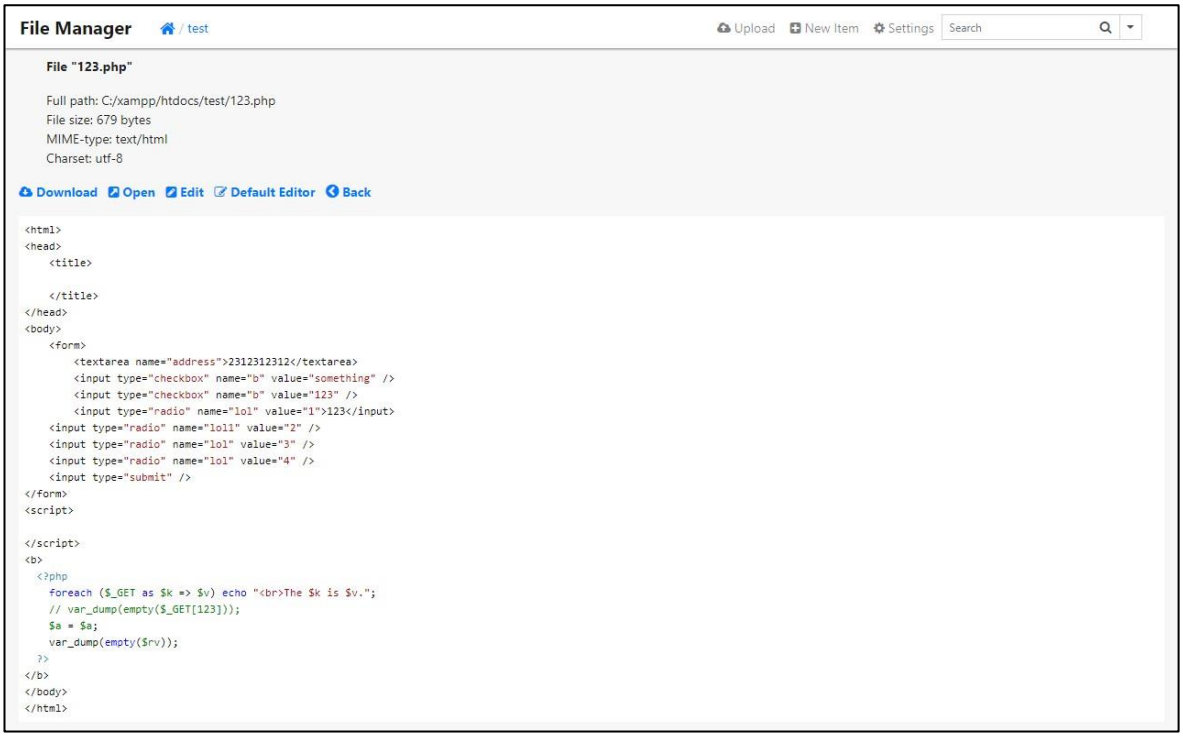


Fig. 6.3.15.4: Viewing the contents of a text file (PHP script)

PHP Interpreter

The PHP Interpreter is basically a (customized) iframe window that comes along with a file editor (default editor) and a tons of other features both for the interpreter and as well as the file manager.

1. Opening the interpreter screen

One can open the interpreter screen for a given editable file by viewing it (clicking on it) and choosing the 'Default Editor' option. (If it is not present then it means that the file cannot be edited). This shall popup the interpreter screen.

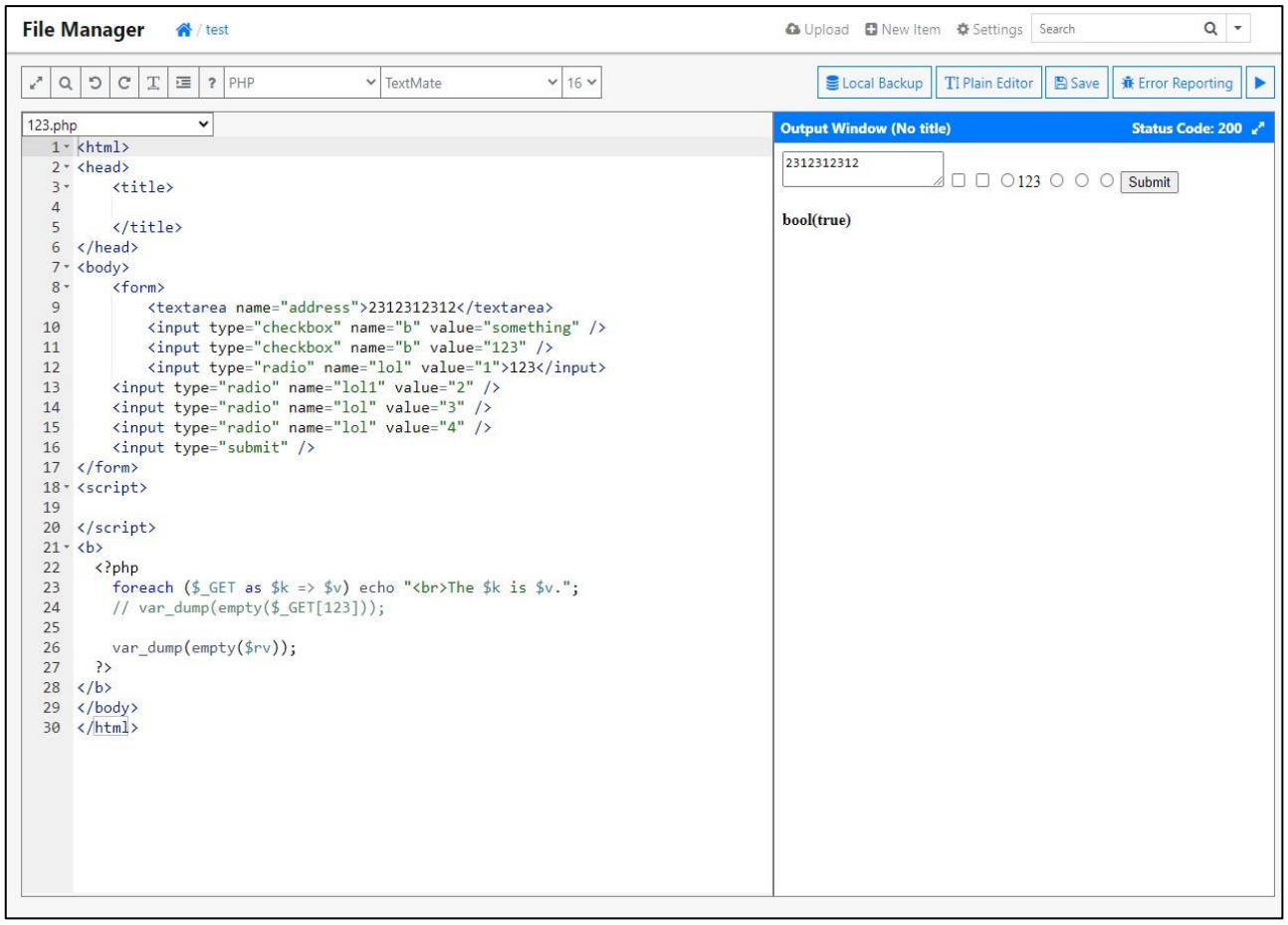


Fig. 6.3.16: Editing a PHP file with the help of the default editor

2. Viewing the editor in full screen

One can easily view the editor in full screen by clicking on the expand icon on top of the editor.

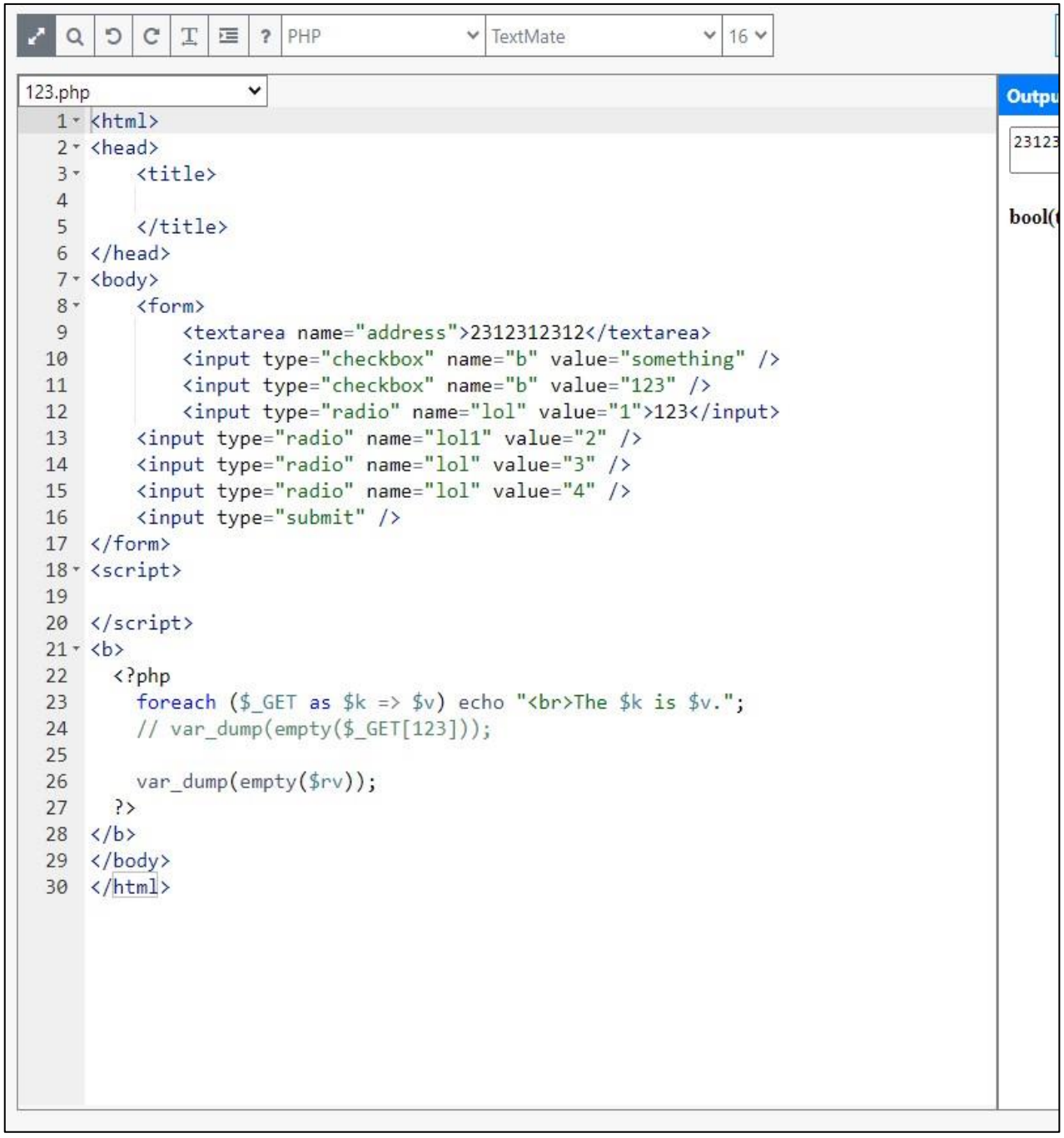


Fig 17.1: An image highlighting the option to be selected for full screen menu

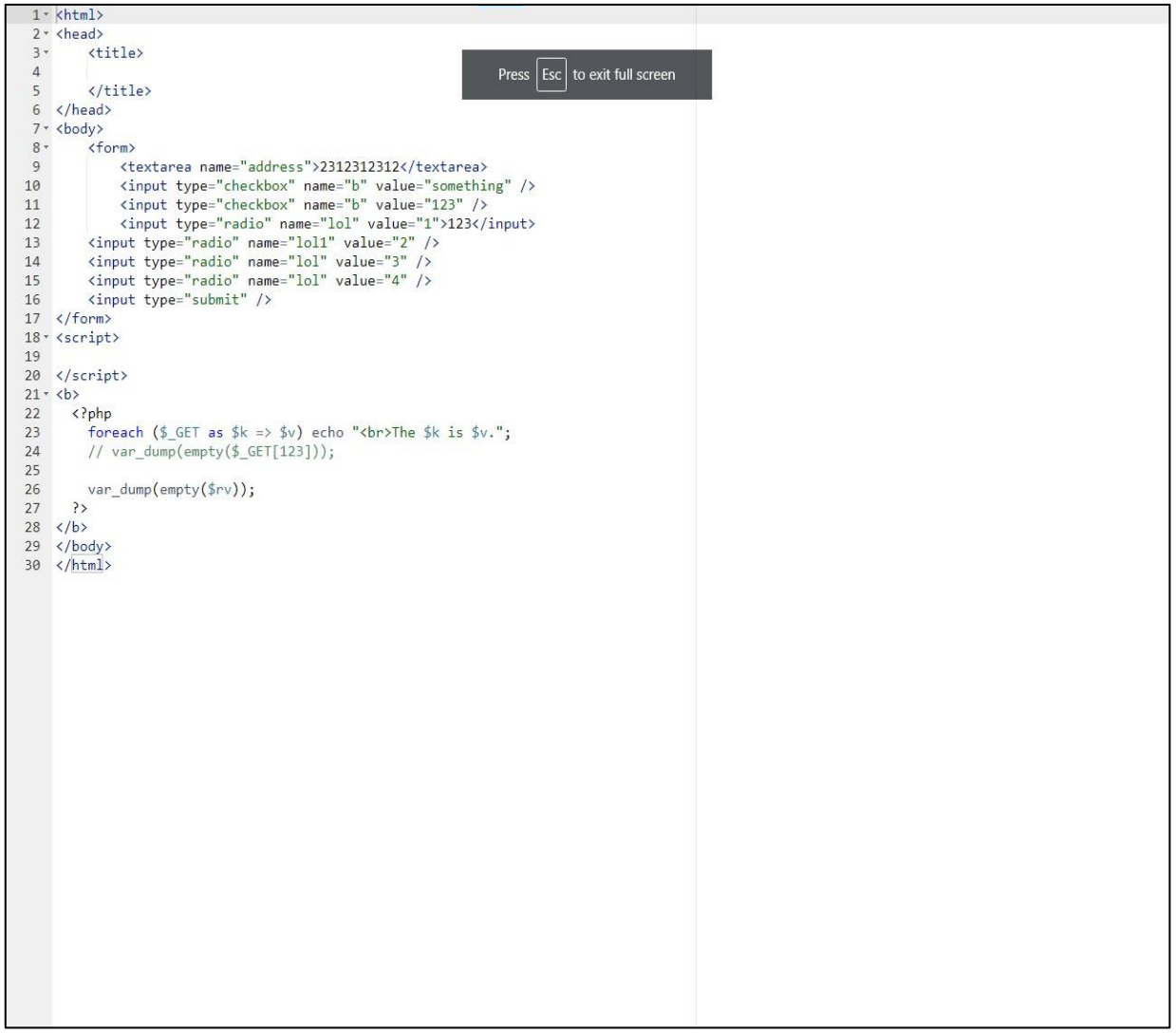


Fig 17.1: The editor in full screen mode

3. Looking for/Replacing a word/words in a file

One can access the 'Find' option either by pressing **Ctrl+F** or by selecting the Search Icon present above the file editor.

If the user wishes to replace a (range of) word(s) with a word, he could use the **'Replace'** option for that. To open that option, one can either press **Ctrl+H** or click on the plus (+) icon present in the 'Find' Menu and press minus icon (–) that would collapse the 'Replace' menu back to the 'Find' menu.

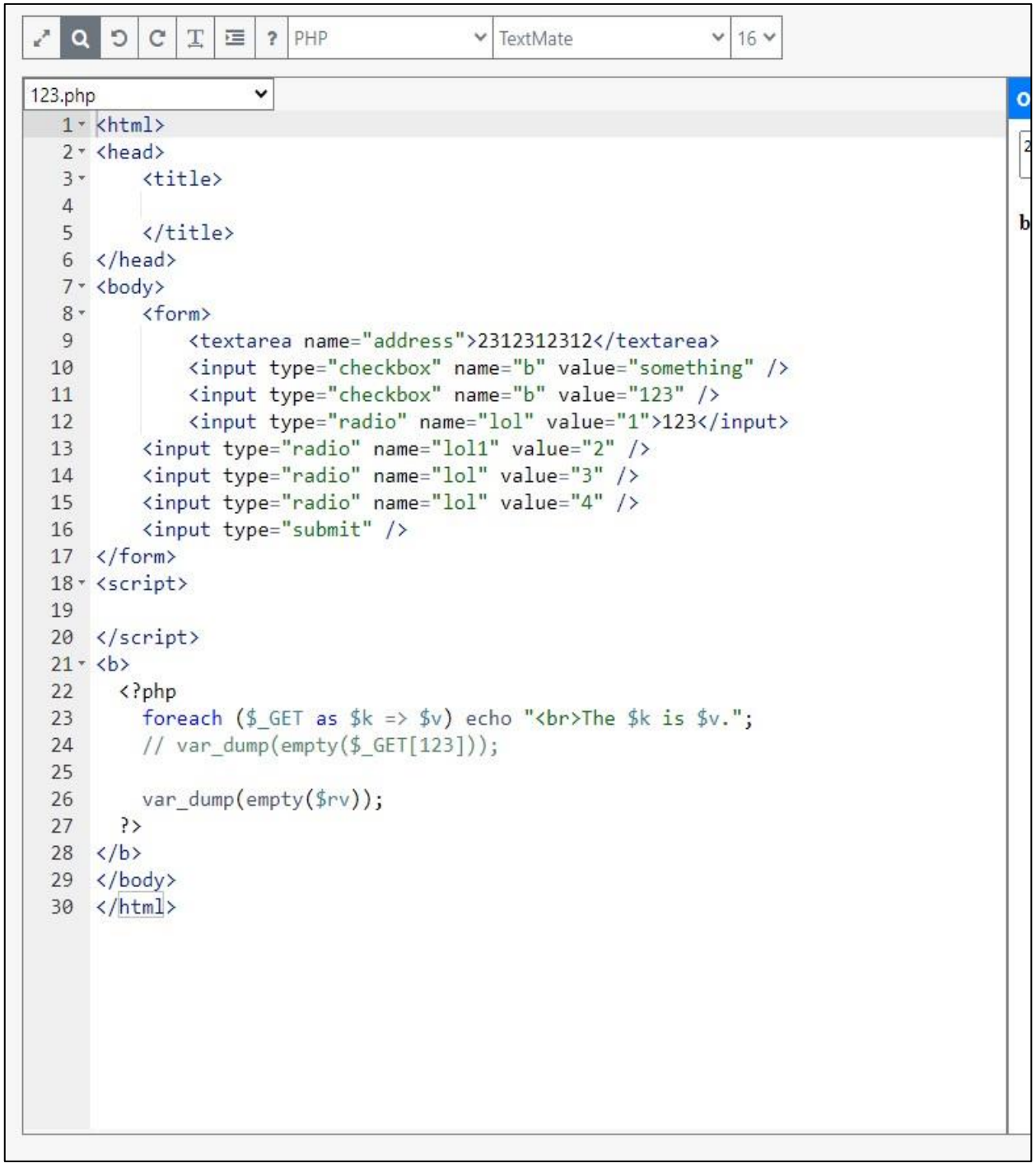


Fig 18: An image highlighting the option to be selected for the ‘Find’ menu

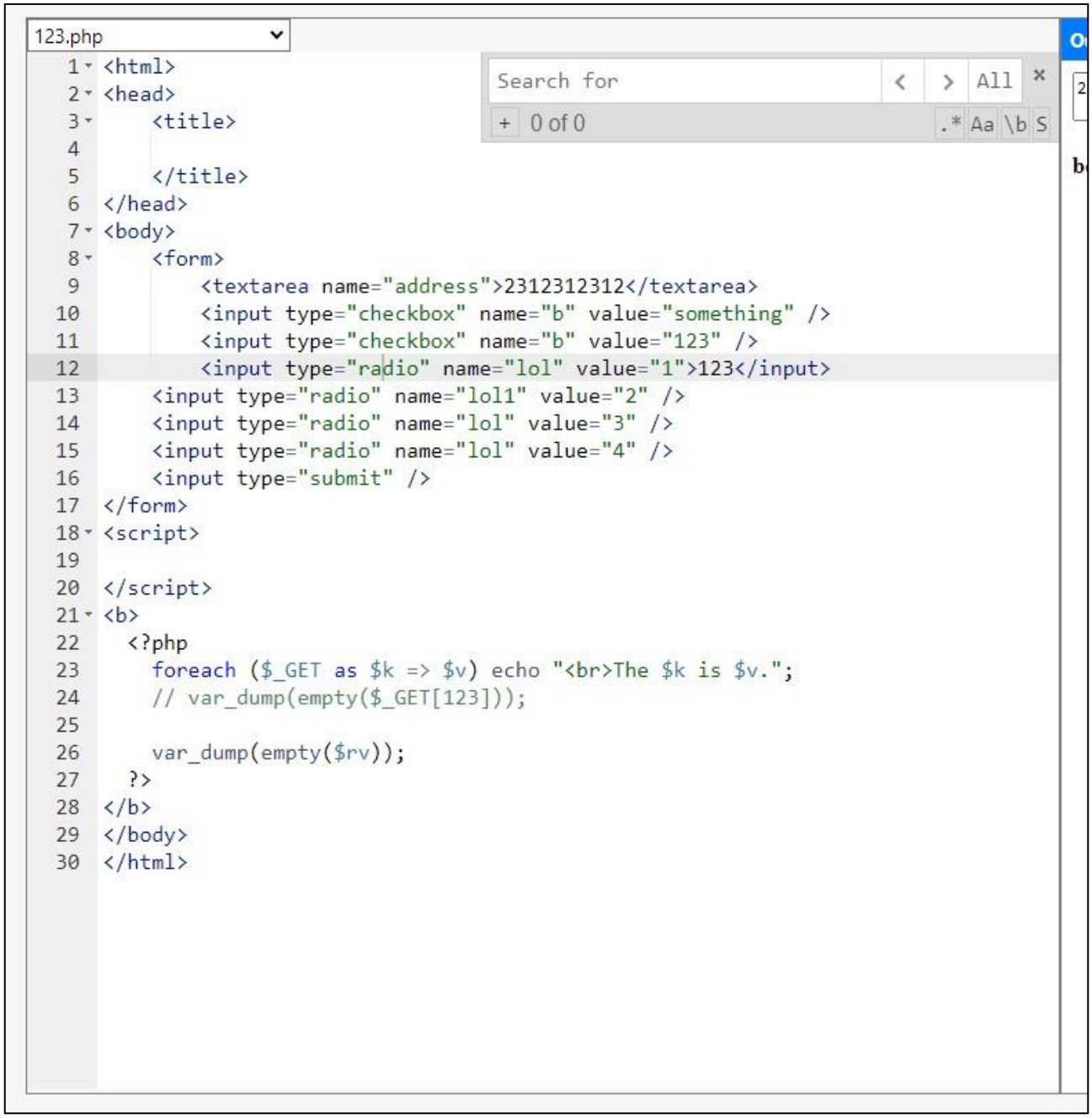


Fig. 6.3.18.2: The 'Find' menu (Top-right corner of the editor)

a. Performing a simple search

By default, the text editor performs an partial text search across all the contents of the file.

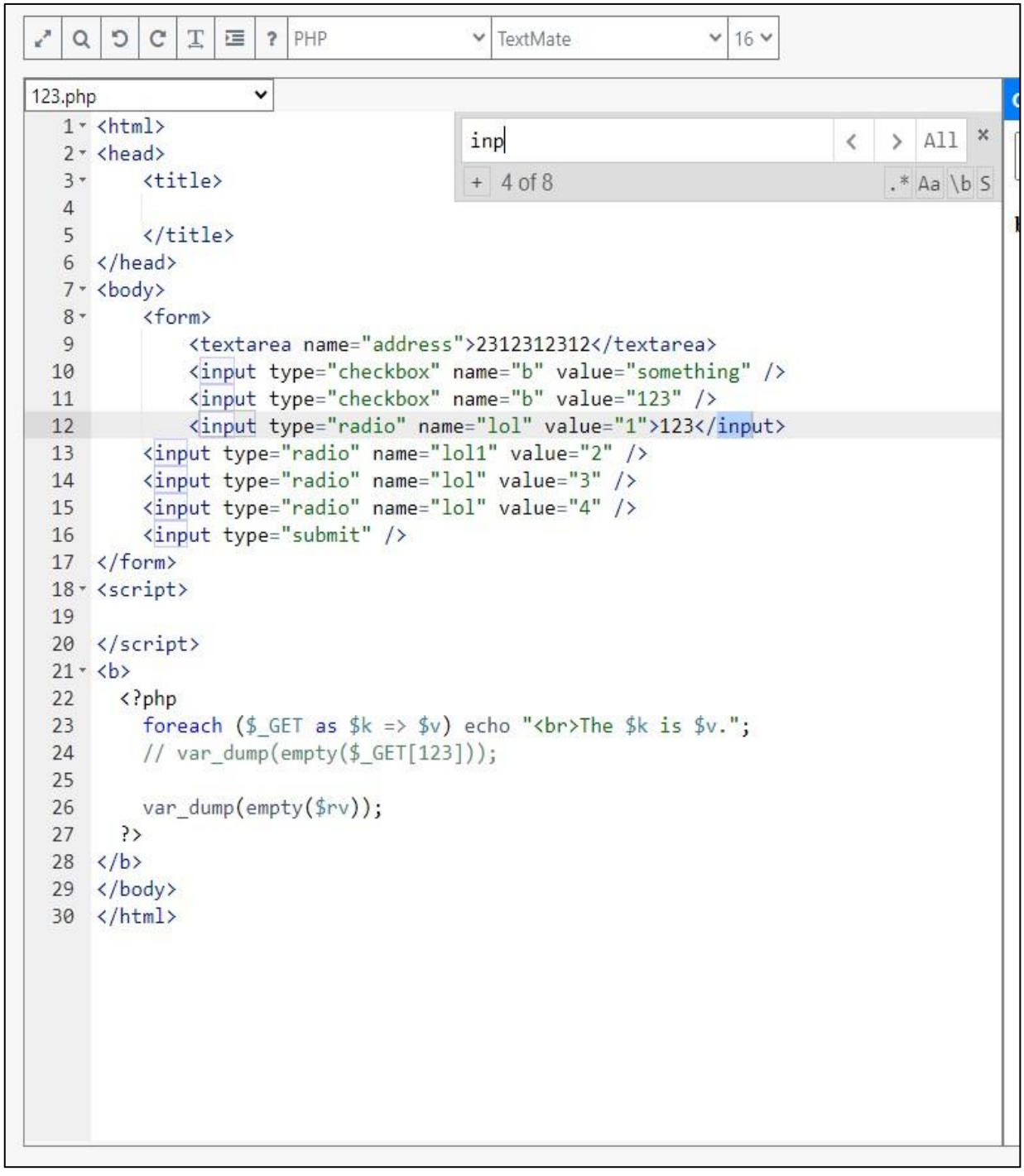


Fig. 6.3.18.a: Performing a simple (partial) text search/Highlighting the found text

b. Performing case sensitive search

One can easily add case sensitivity to the search operation by enabling it from the options of the ‘Find’ menu. (The option is highlighted in the below image [the menu])

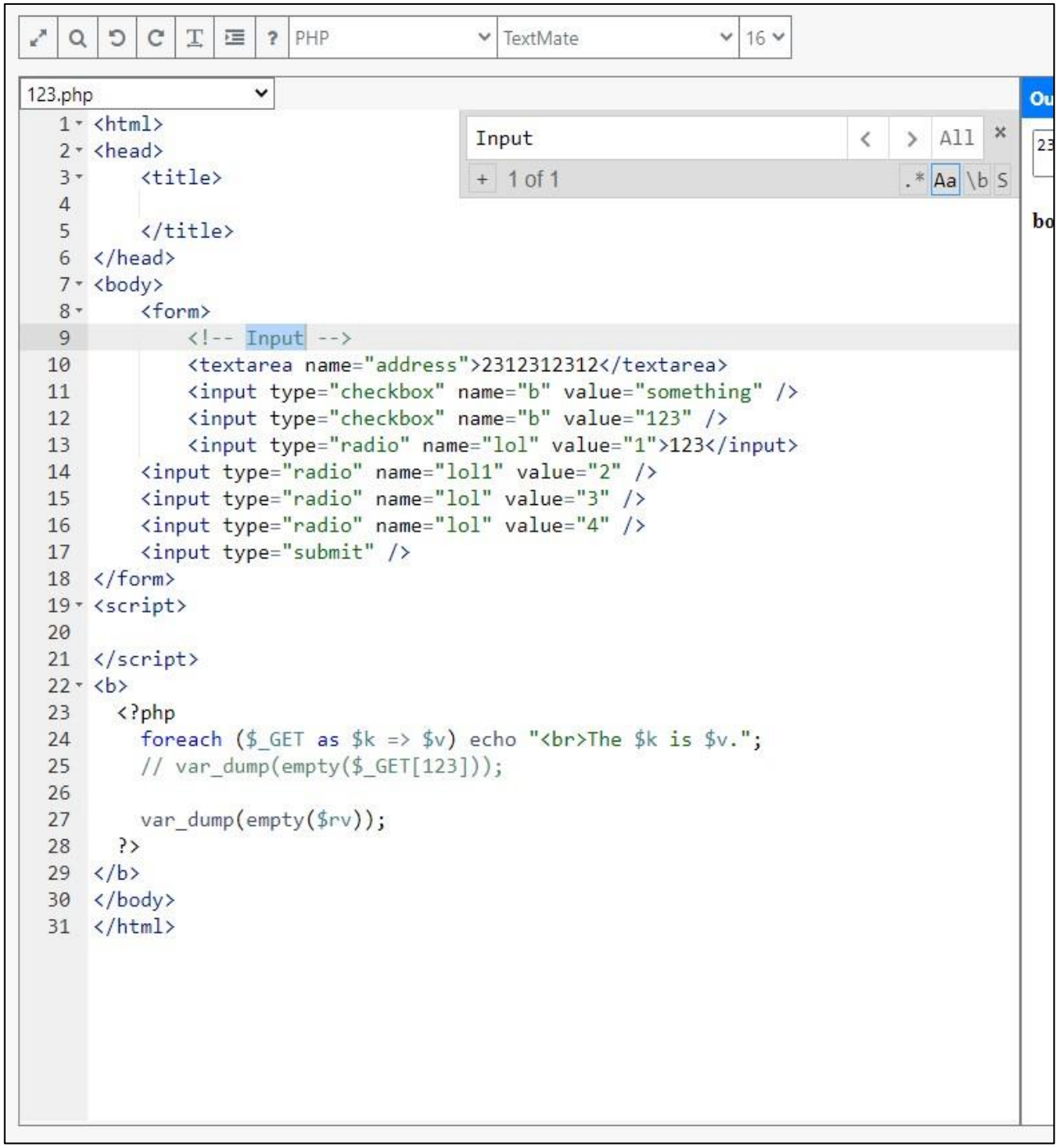


Fig. 6.3.18.b: Performing case-sensitive search operation for the text 'Input'

c. Performing whole word search

One can easily perform whole word search operation by enabling it from the options of the 'Find' menu. (The option is highlighted in the below image [the menu])

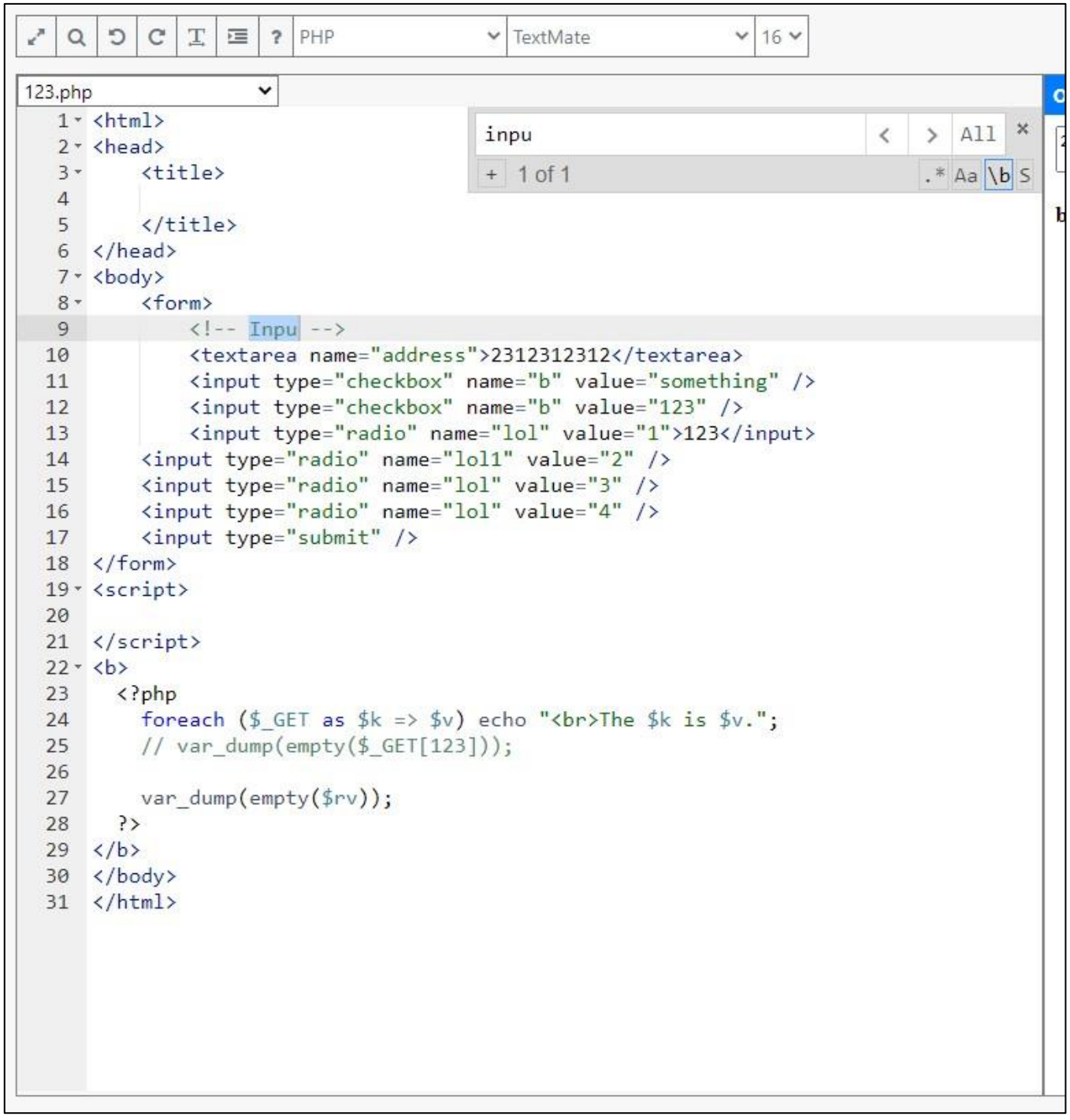


Fig. 6.3.18.b: Performing case-sensitive search operation for the text 'Input'

c. Searching in a given range of selection

One can perform search in a given range only by using the 'Search in Selection' option, that has been highlighted in the below image. (the count of the found items)

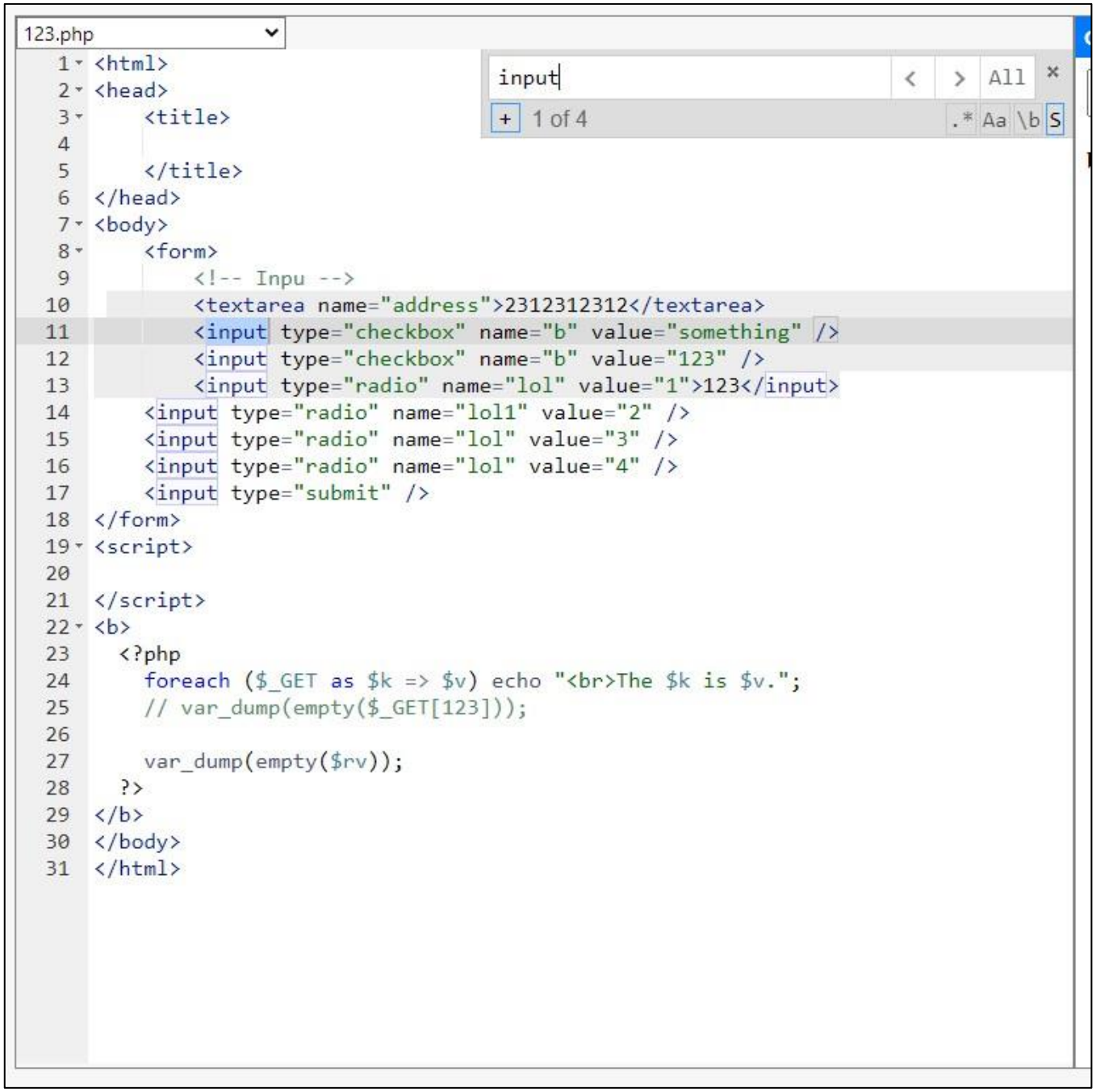


Fig. 6.3.18.c: Performing search within a given selection

d. Searching using regular expressions (regex)

One can easily search for texts matching a given pattern using the ‘Regex’ option available in the ‘Find’ menu (the option has been highlighted in the below image).

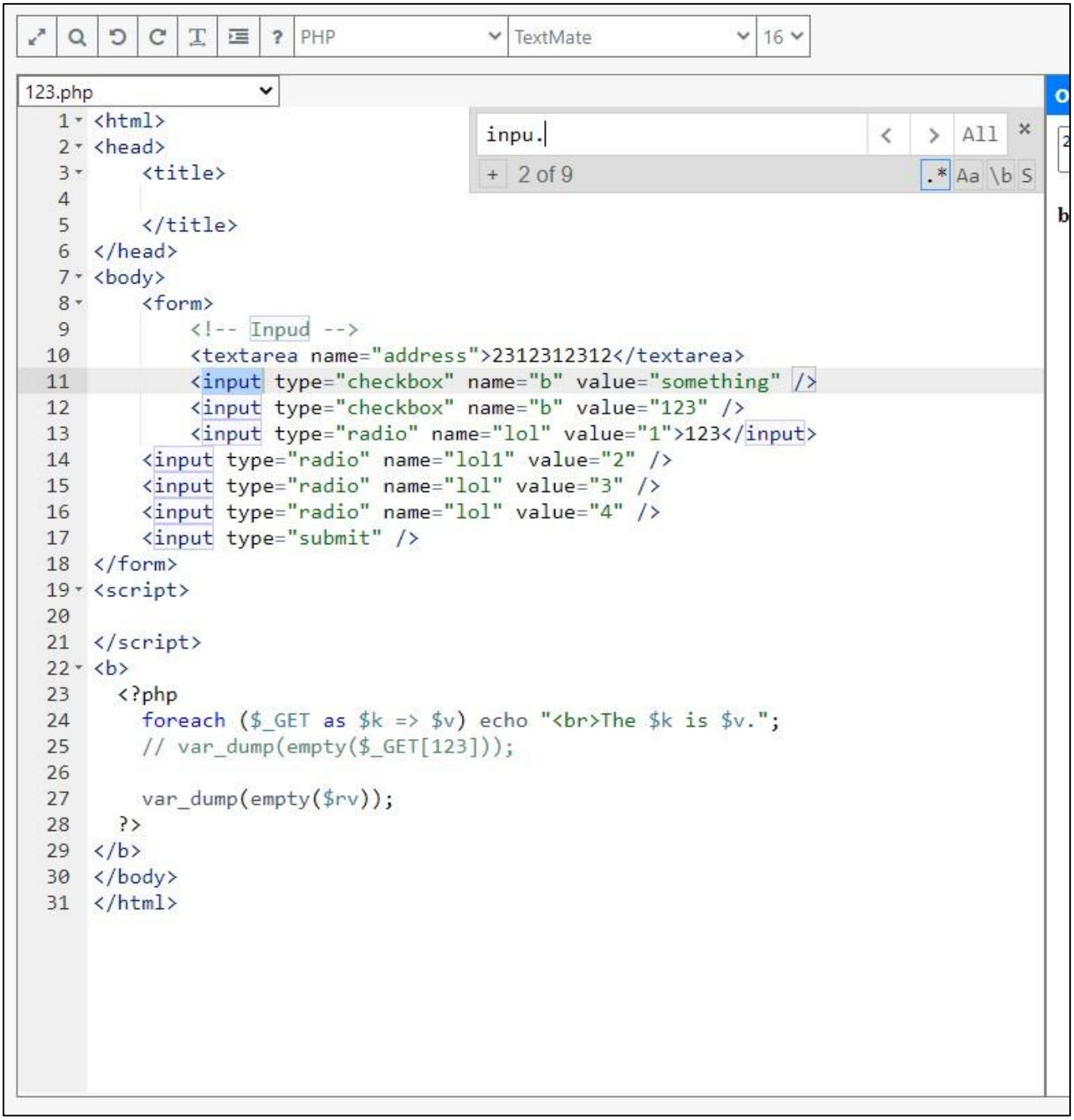


Fig. 6.3.18.d: Looking for all the words starting with ‘inpu’ with an extra character

e. Replacing text with the help of ‘Replace’ menu

With the help of the ‘Find’ part of the menu one can look for instances that the user desires to replace, and with ‘Replace’ enter the fixed text that needs to be replaced.

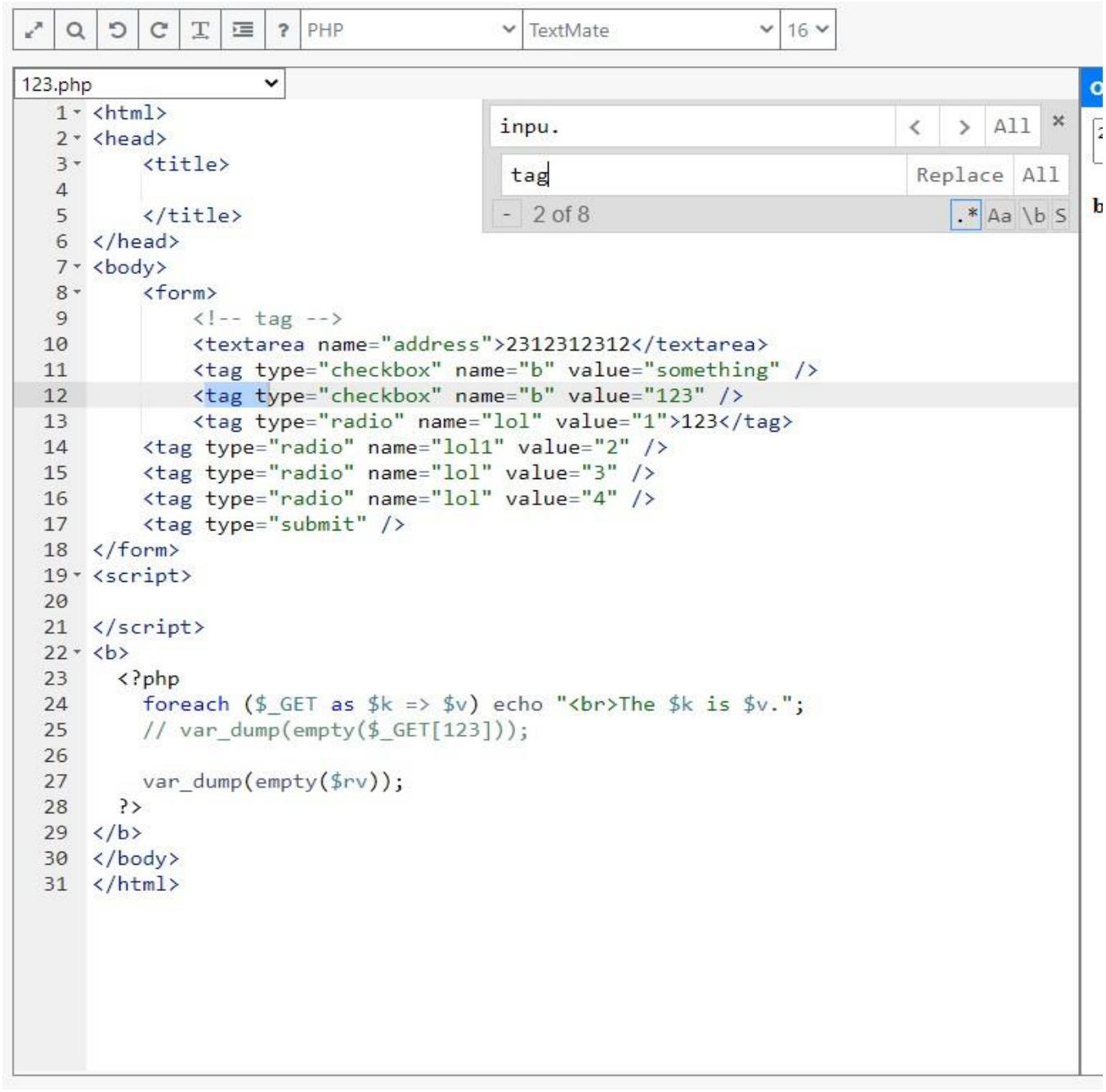


Fig. 6.3.18.e: Replacing all the found matches with the fixed text ‘tag’

4. Undo-ing a change made in the current session

One can easily undo a change that the user had previously made by either clicking the Undo icon (highlighted in the below image) above the editor or by pressing **Ctrl+Z**.

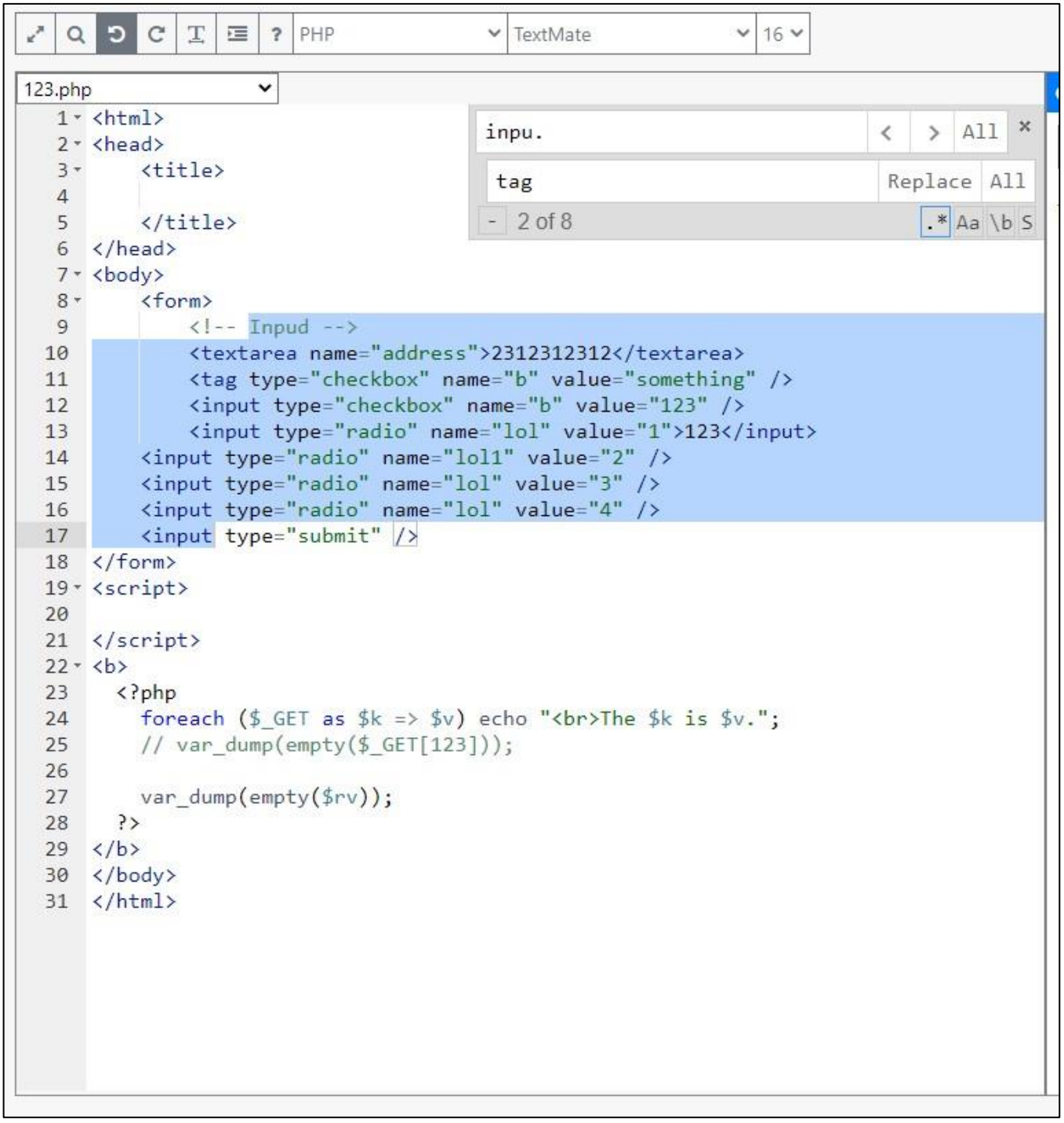


Fig. 6.3.19: Undo-ing a change that was previously made (the last example)

5. Redoing an un-done change

One can easily redo an un-done change by either clicking the Redo icon (highlighted in the below image) above the editor or by pressing **Ctrl+Y** or **Ctrl+Shift+Z**.

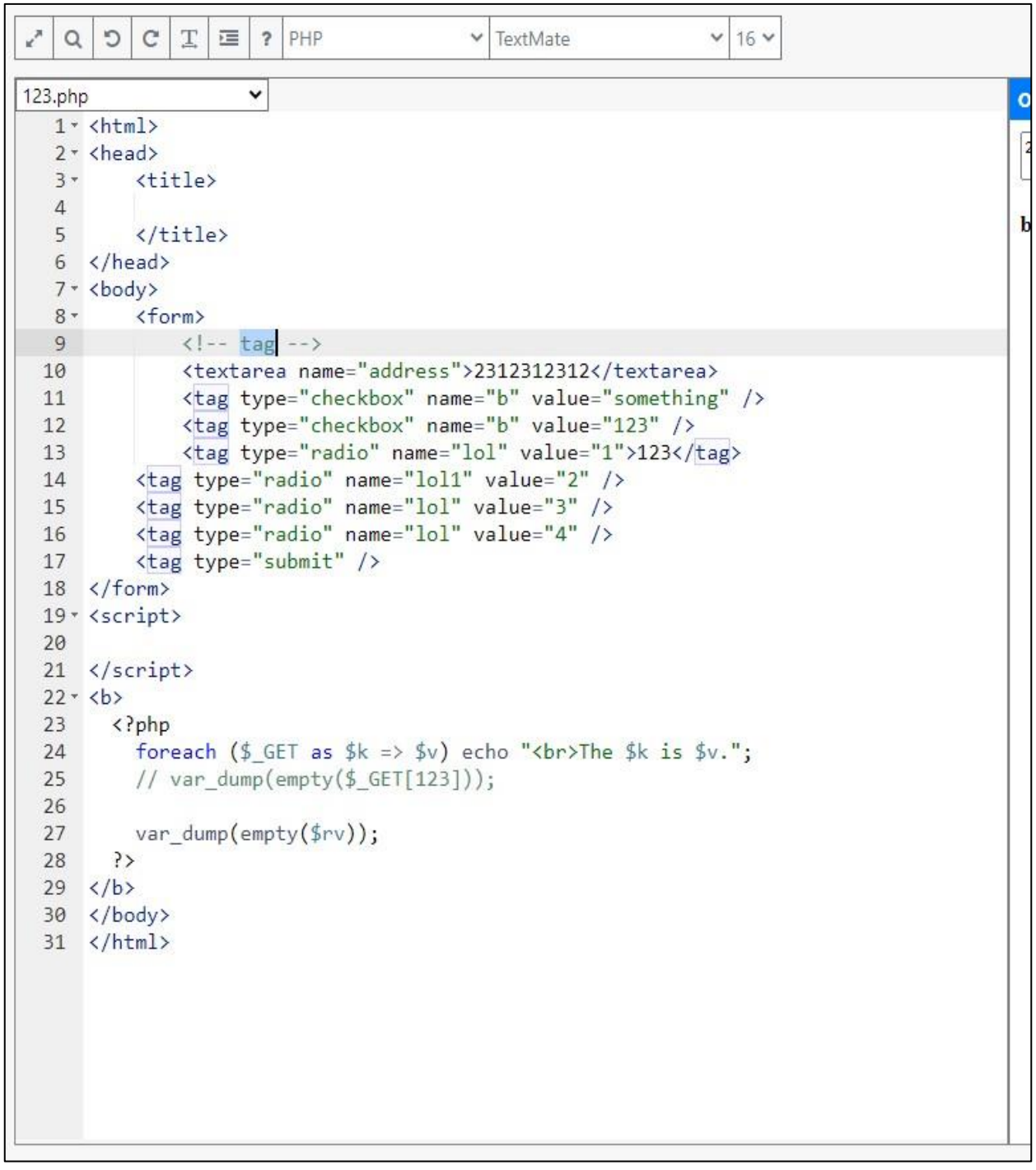


Fig. 6.3.20: Redoing an undone change

6. Resizing the file editor and output window

The editor and output window can be resized relative to each other by dragging the line that separates both (abstract).

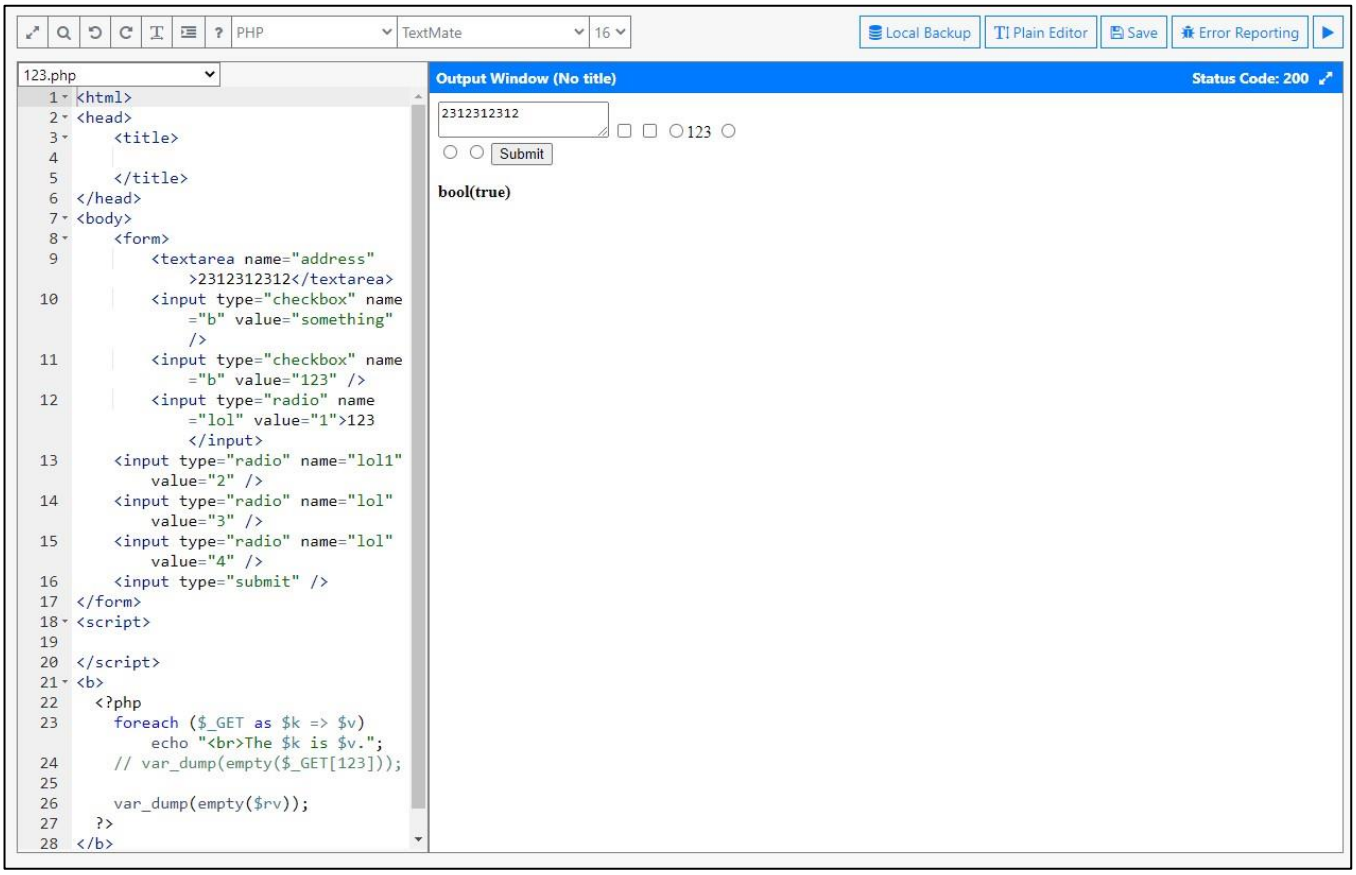


Fig. 6.3.21: Resizing the editor/output window

7. Word Wrap

A user can word wrap all the text based on the current width of the of the editor with the help of the 'Word Wrap' option. One can perform Word Wrap by clicking the word wrap icon above the editor (highlighted in the next image).

The editor automatically tries to word wrap when the editor/output window gets resized, however in some cases the browser might not detect the completion of the resize event. So in those cases one might need to manually word wrap the contents after resizing. So this option might help.

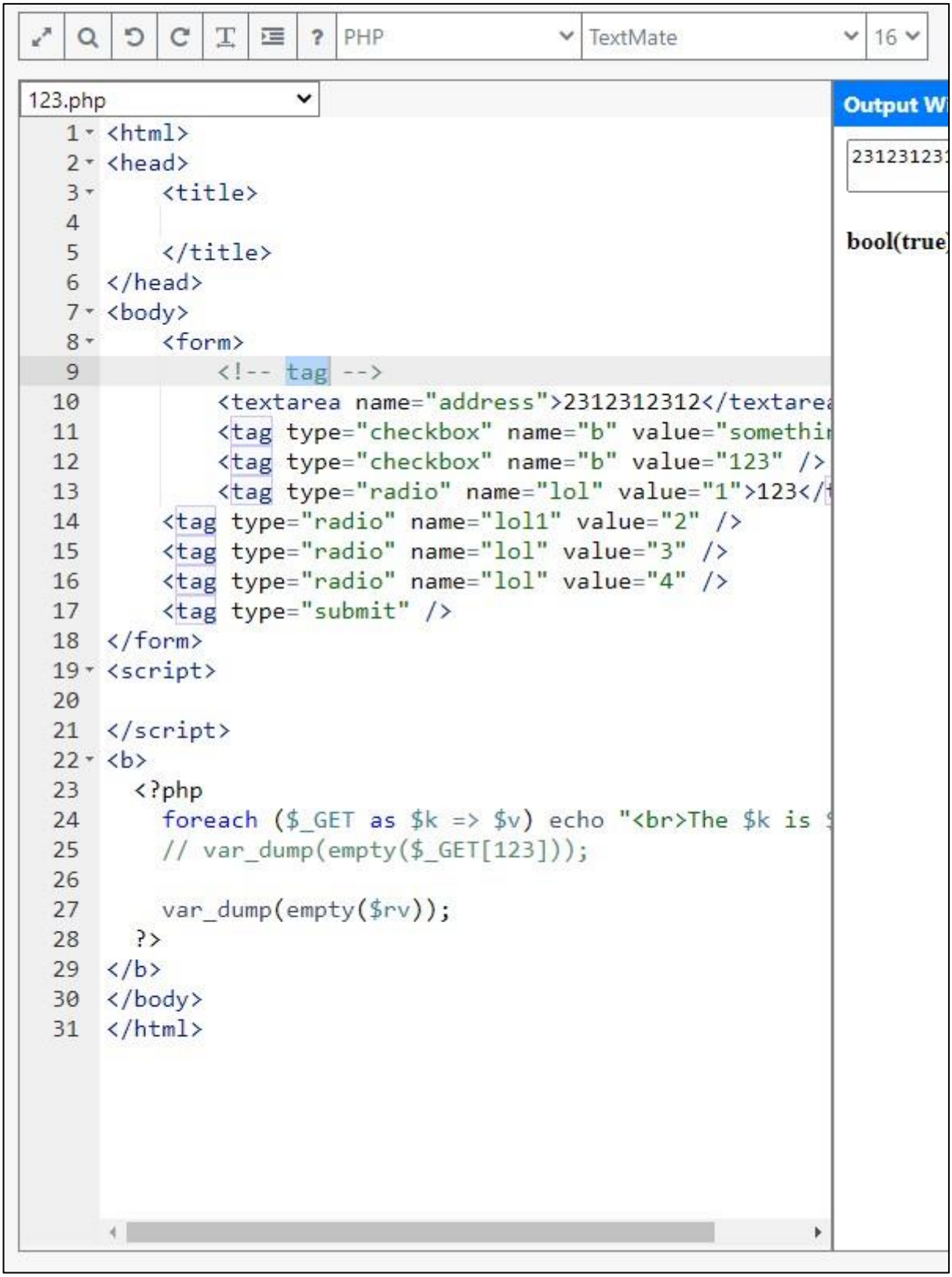


Fig. 6.3.21.1: Before Word Wrap

8.

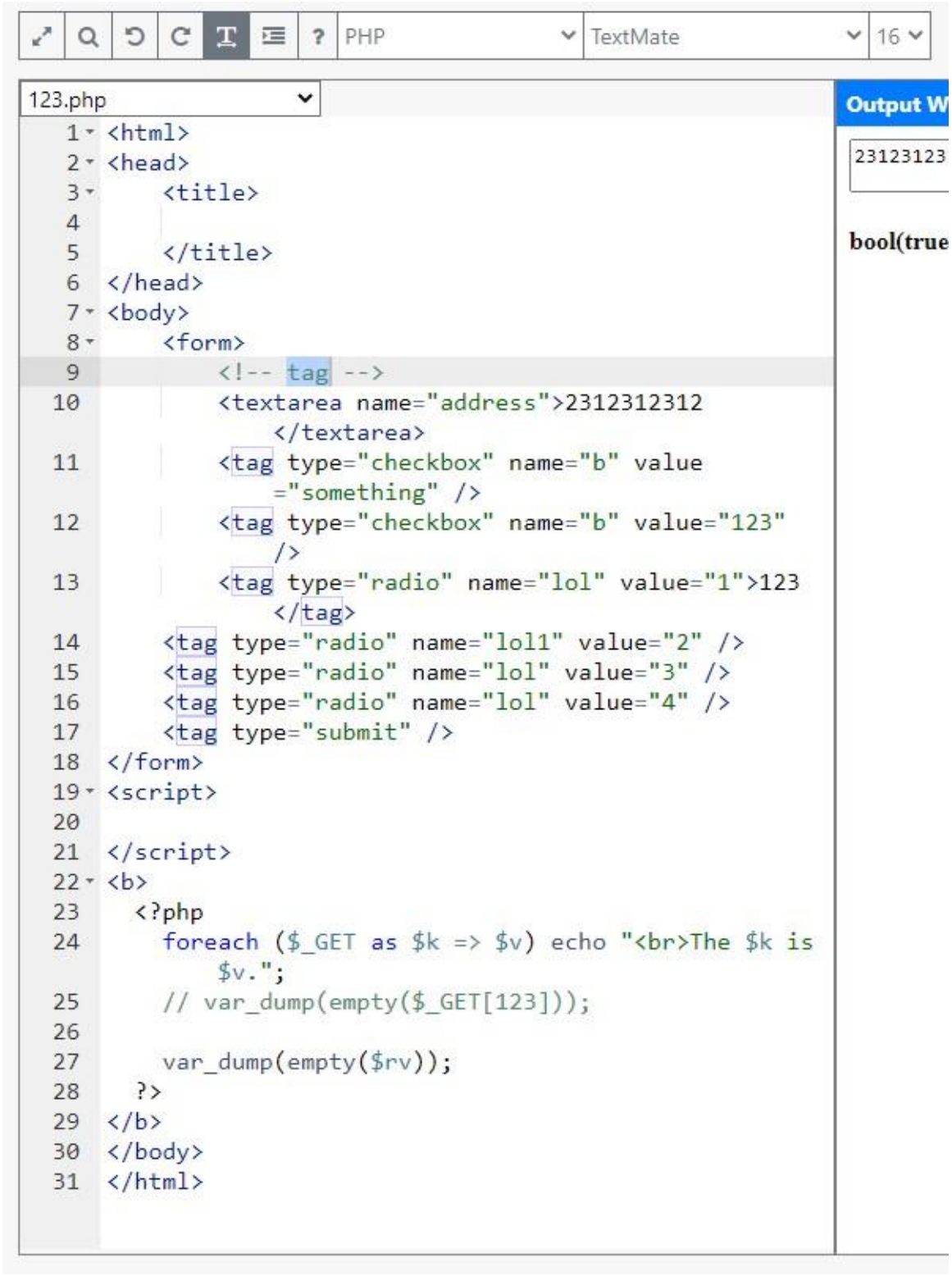


Fig. 6.3.21.2: After Word Wrap

Formatting/Beautifying the code

One could format his code as well with the help of the Beautify option. This will basically format the current file on the basis of the file extension and the contents of the file (based on the primary standards of the language). The editor has support for quite a many languages

(100+ languages). This can save time that would otherwise be needed for formatting the code.

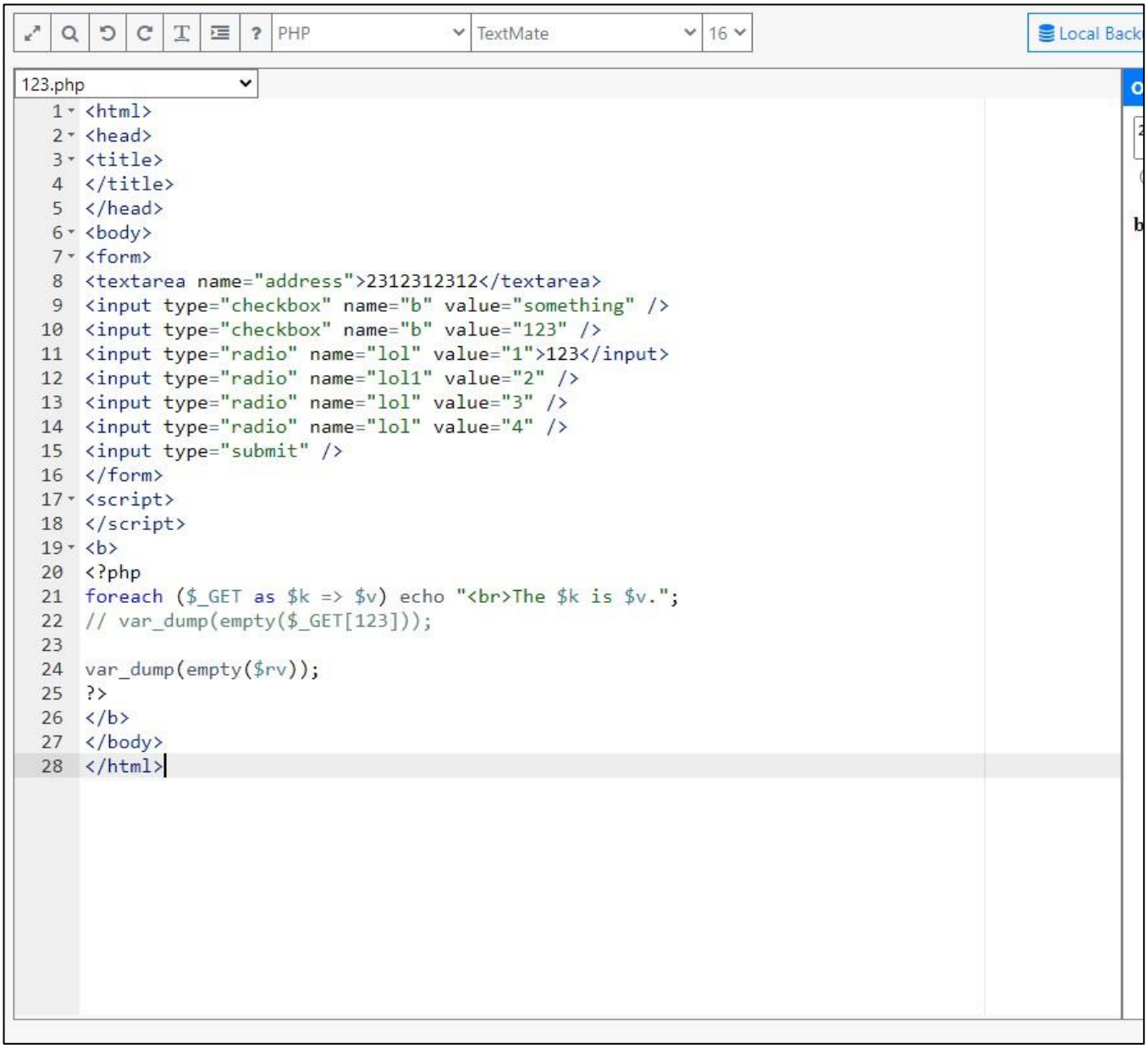


Fig. 6.3.22.1: Before Formatting

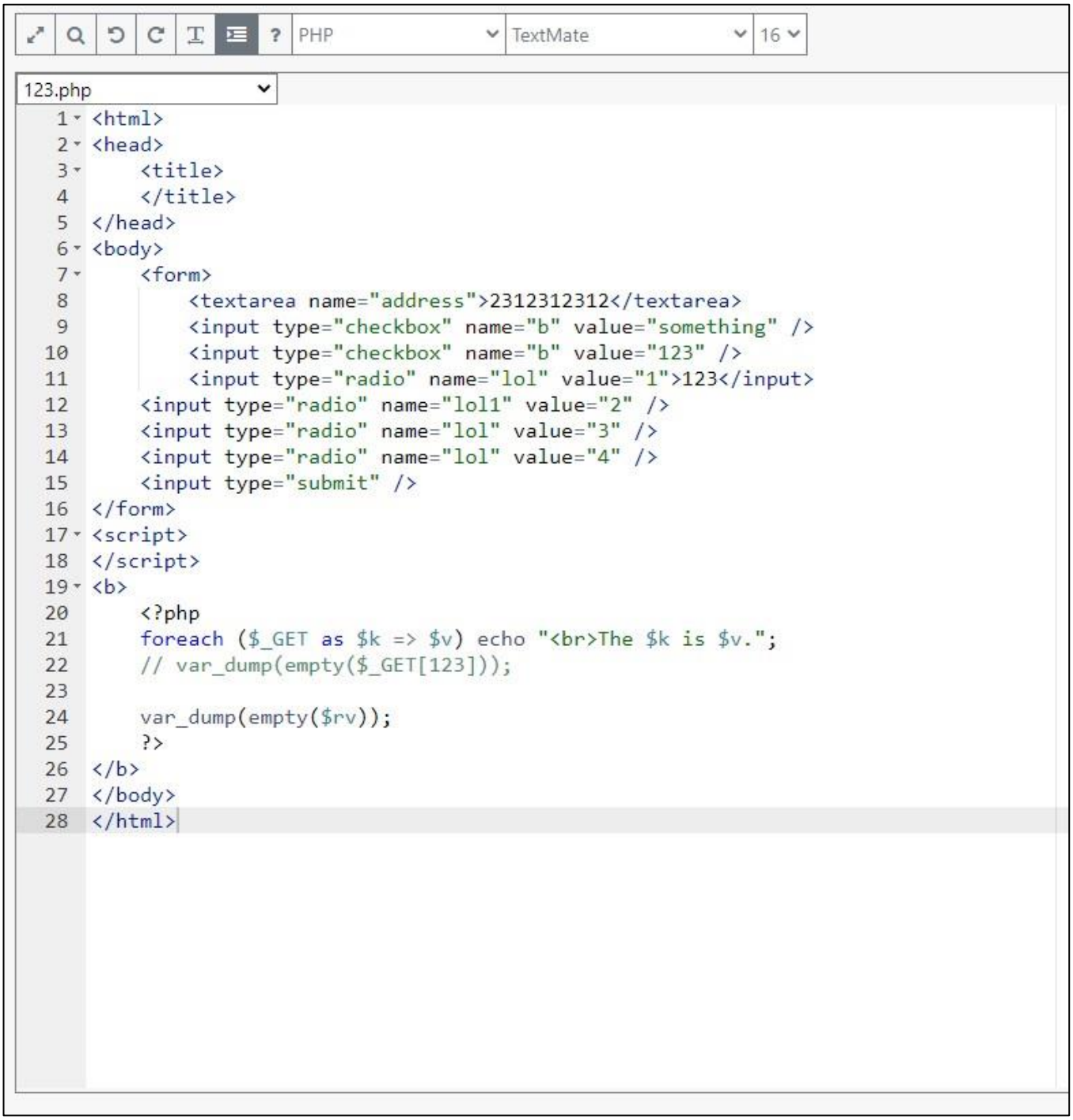


Fig. 6.3.22.2: After Formatting

9. Displaying the help menu

A user can view the ‘Help’ menu by clicking the ‘Help’ icon above the file editor.

Fig 23.1: An image where the help icon has been highlighted

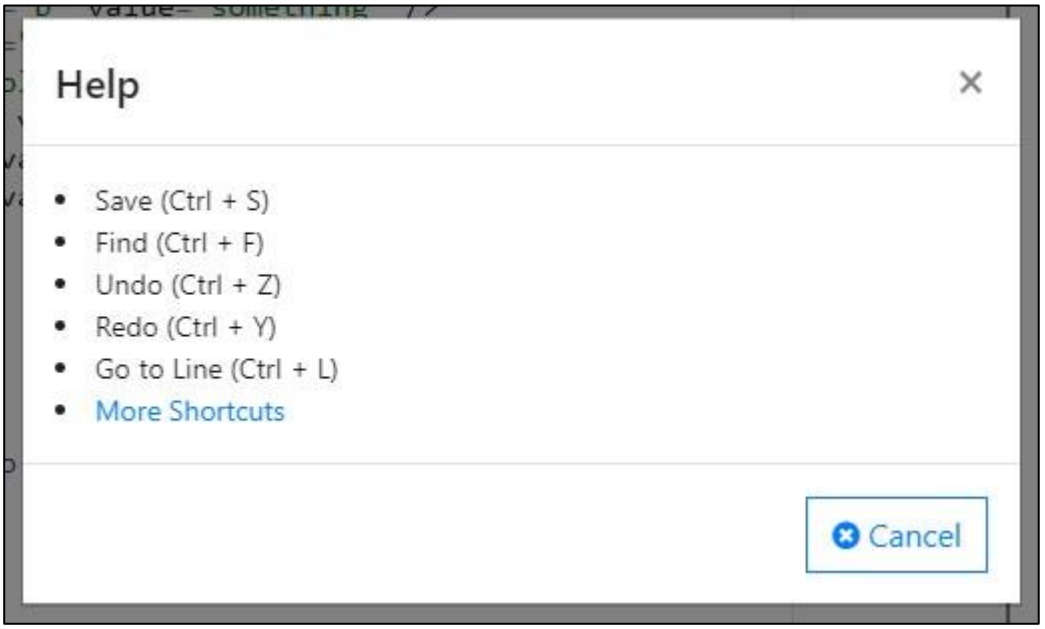


Fig 23.2: The help menu

10. Changing the language template and theme of the editor

The language that is being assumed by the editor or the theme of the editor can be changed by changing it from the dropdowns available above the editor.

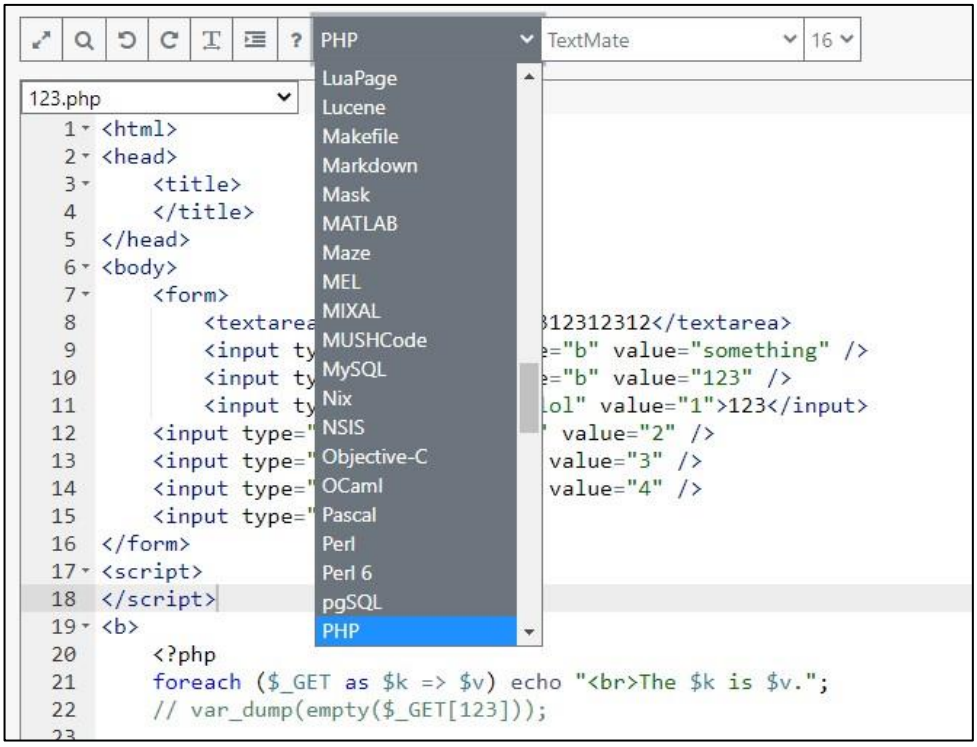


Fig. 6.3.24.1: The dropdown menu for changing the language template

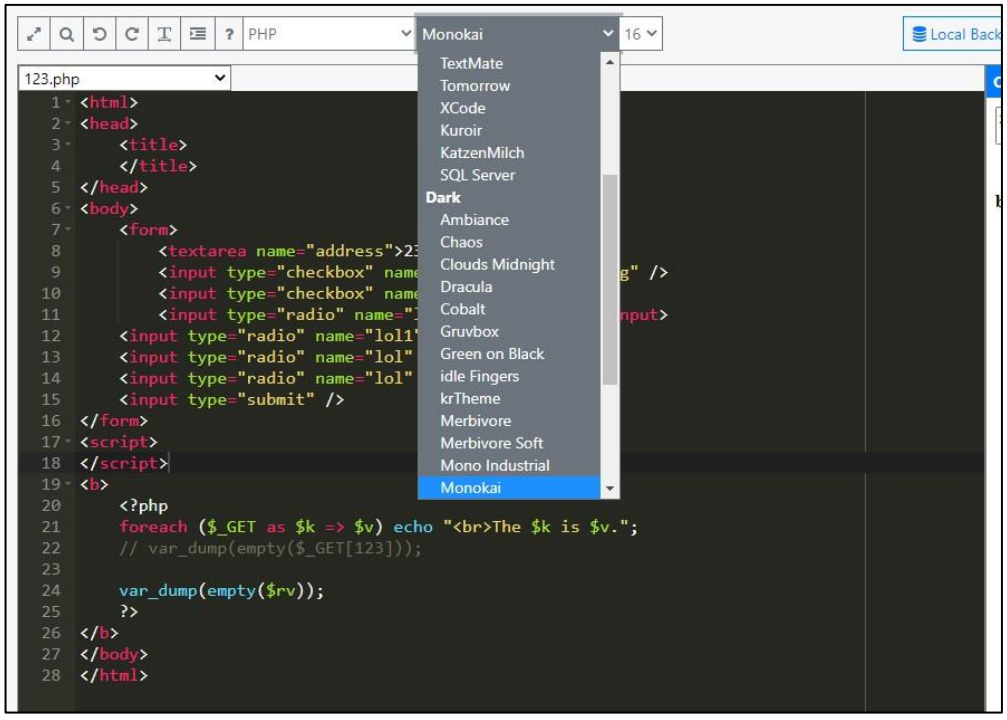


Fig. 6.3.24.2: The dropdown menu for changing the theme of the editor

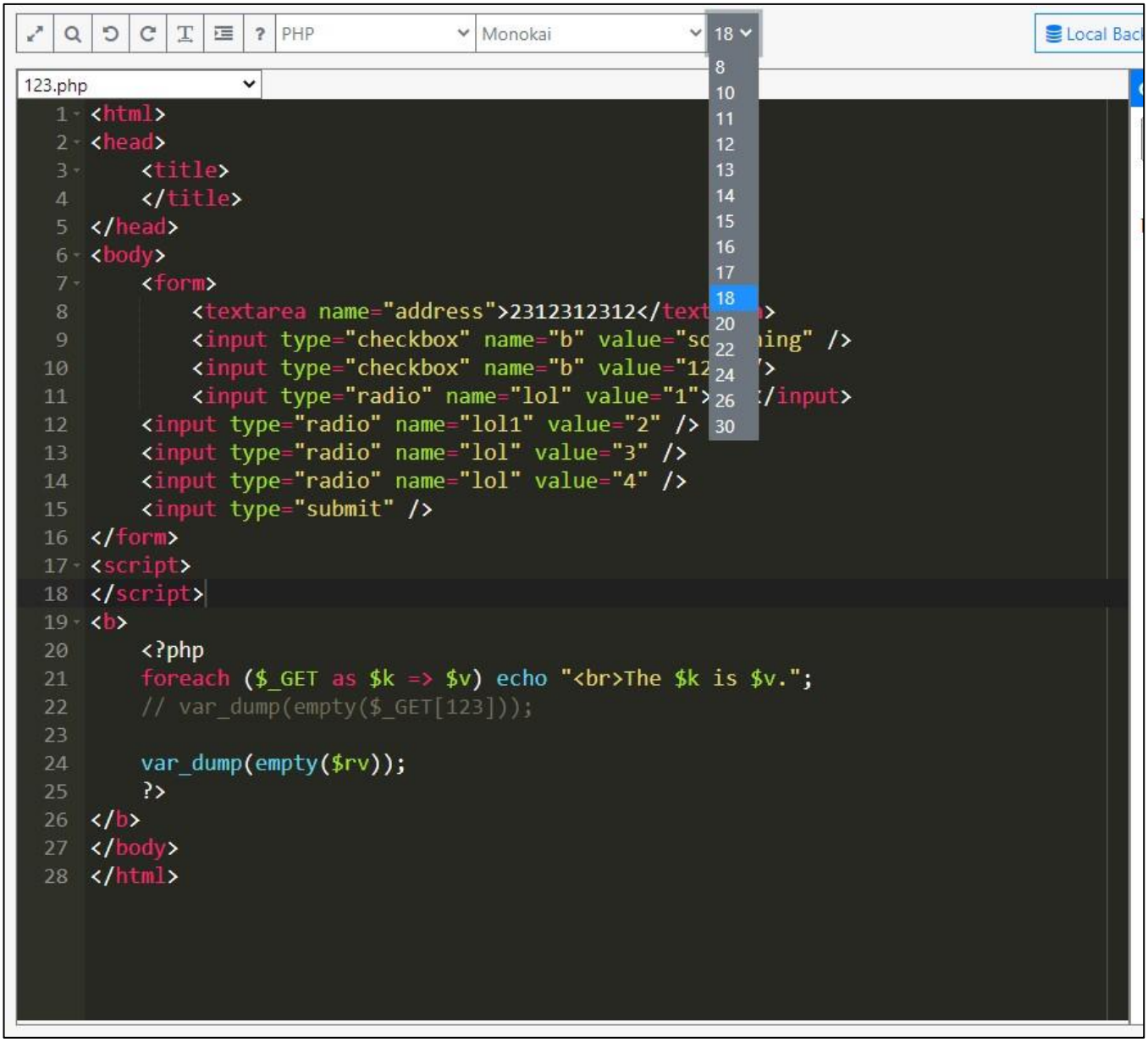


Fig. 6.3.24.3: Changing the font size of the editor from 16 to 18

11. Switching files within the same directory

One can quickly change files within the same directory by using the dropdown that is present over the editor. It not only adds to the user's convenience but also as well saves time.

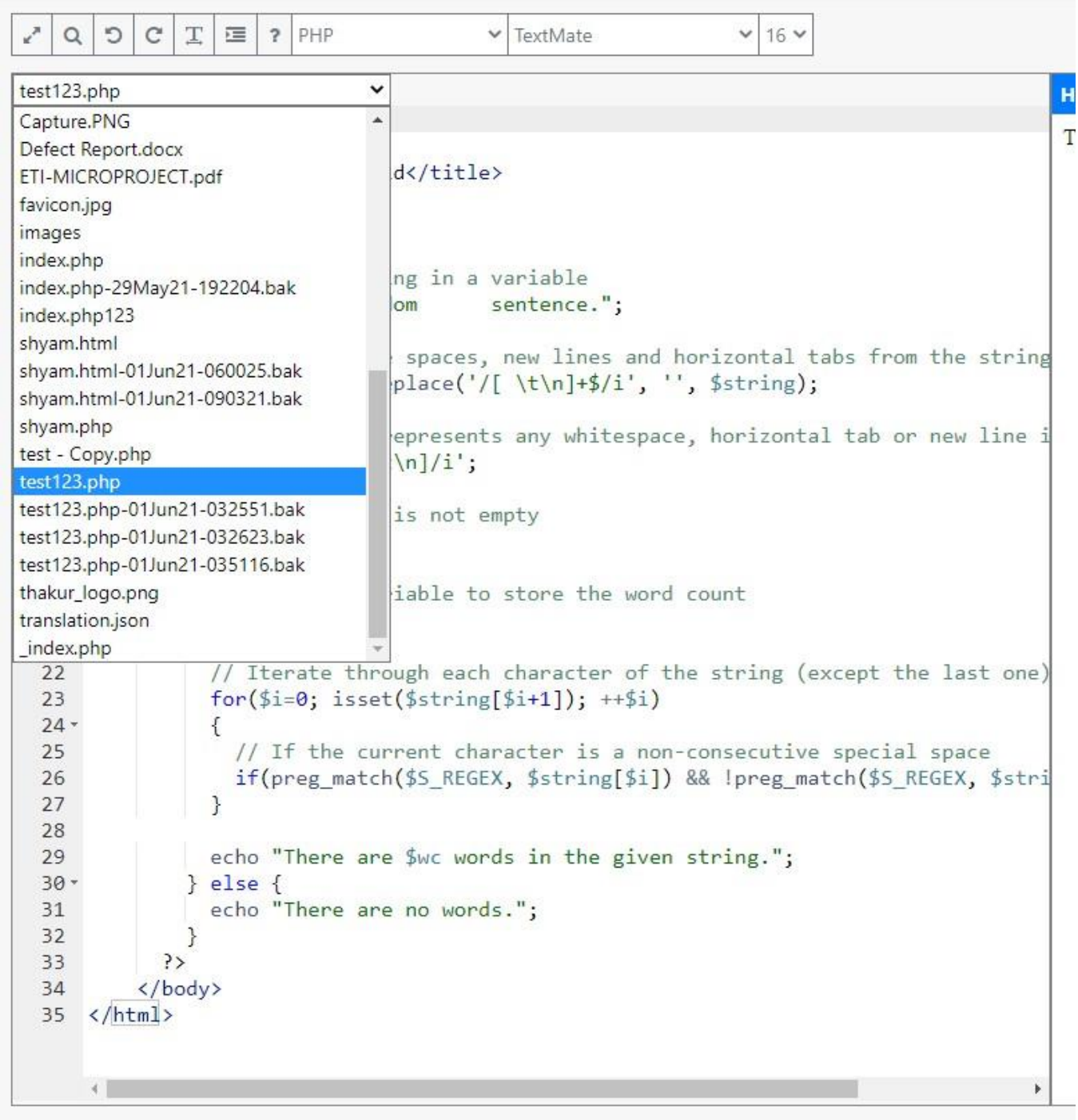


Fig. 6.3.25: Switching files within the same directory

12. The output window

The output window views the file being viewed in its actual form (as it would be viewed on a browser). In the header, the left hand side has the title of the page being displayed and the right hand side has the status code (for debugging/testing purposes).

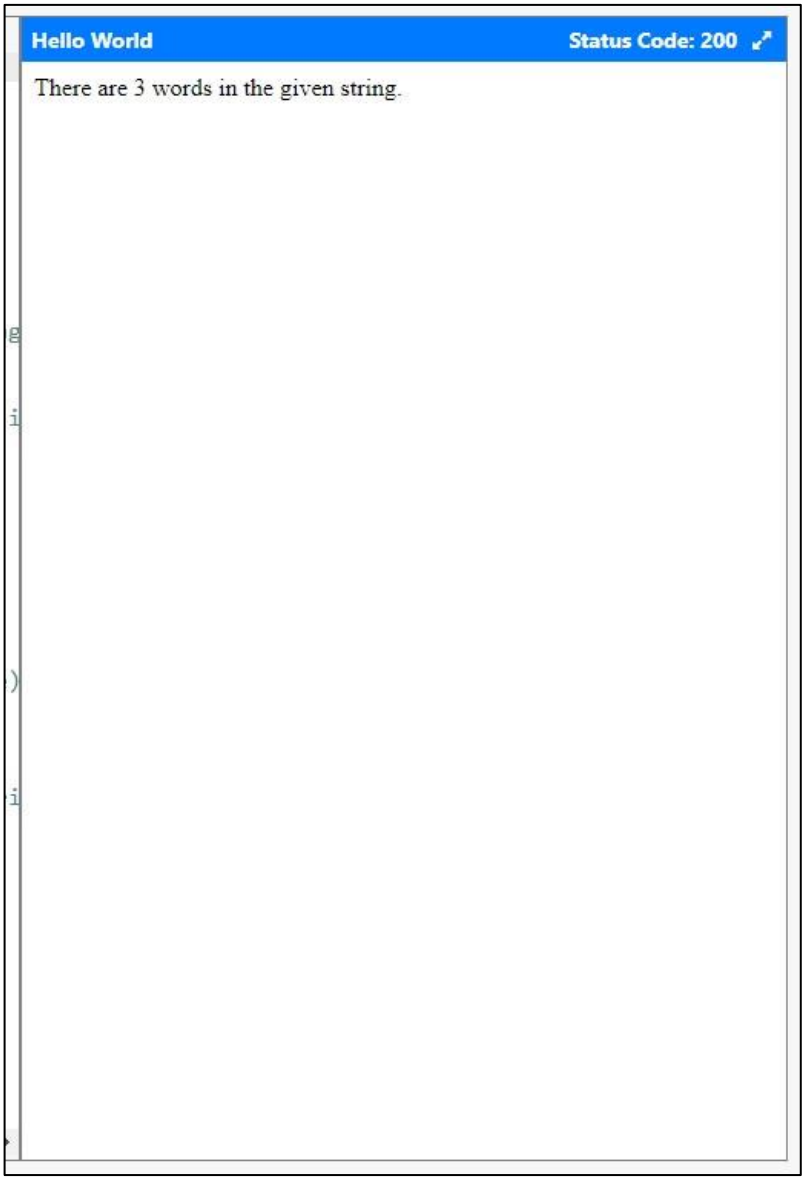


Fig. 6.3.26.1: The output window

Along with that there is an option to view the file in full screen mode.

Fig. 6.3.26.2: The output window (in full screen mode)

Sometimes the server/page might send an invalid status code that might cause the output window to completely crash. In those cases, only reloading the page is an option.

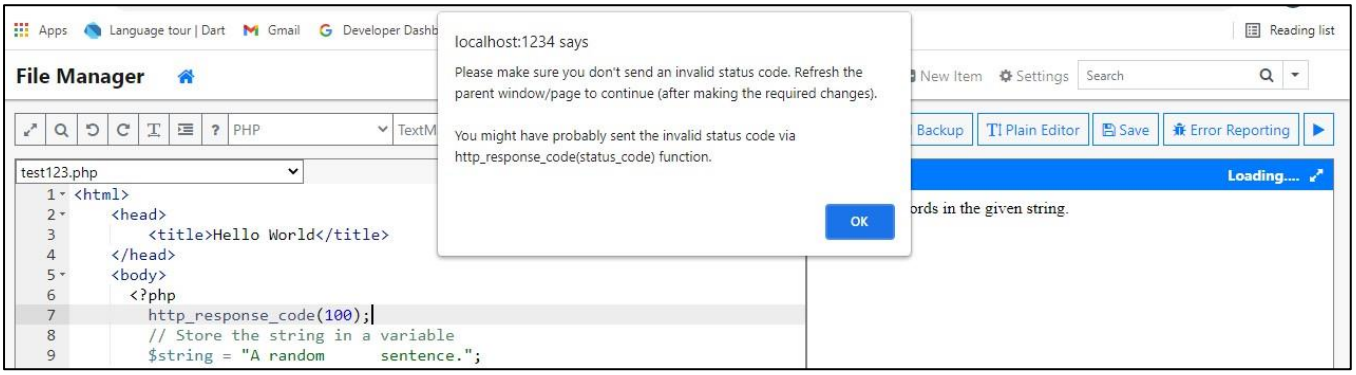


Fig. 6.3.26.2: The alert dialog that appears before the window crashes (due to invalid status code)

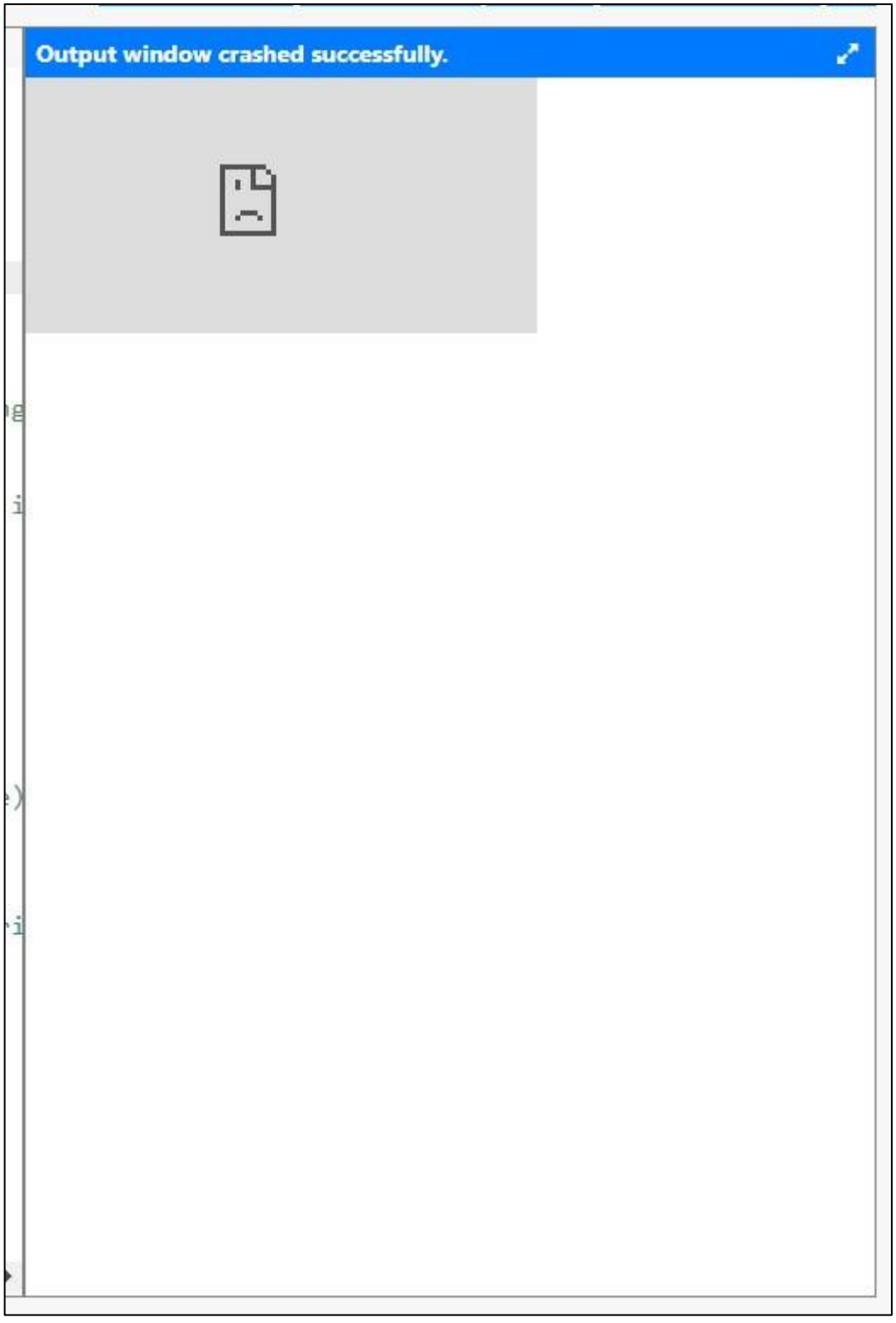


Fig. 6.3.26.3: A crashed output window example (due to invalid status code)

13. Taking local backup

There are times when a developer would want to save their file at a specific point of time so that they can revert back to it in the future, so for those cases the developer can the current instance of the file by clicking on the **Local Backup** option.

It will basically save a file with following name format as a backup along with the contents of the current file in the same directory:

{filename}.{extension}-{DD}{MMM}{YY}-{timestamp}.bak

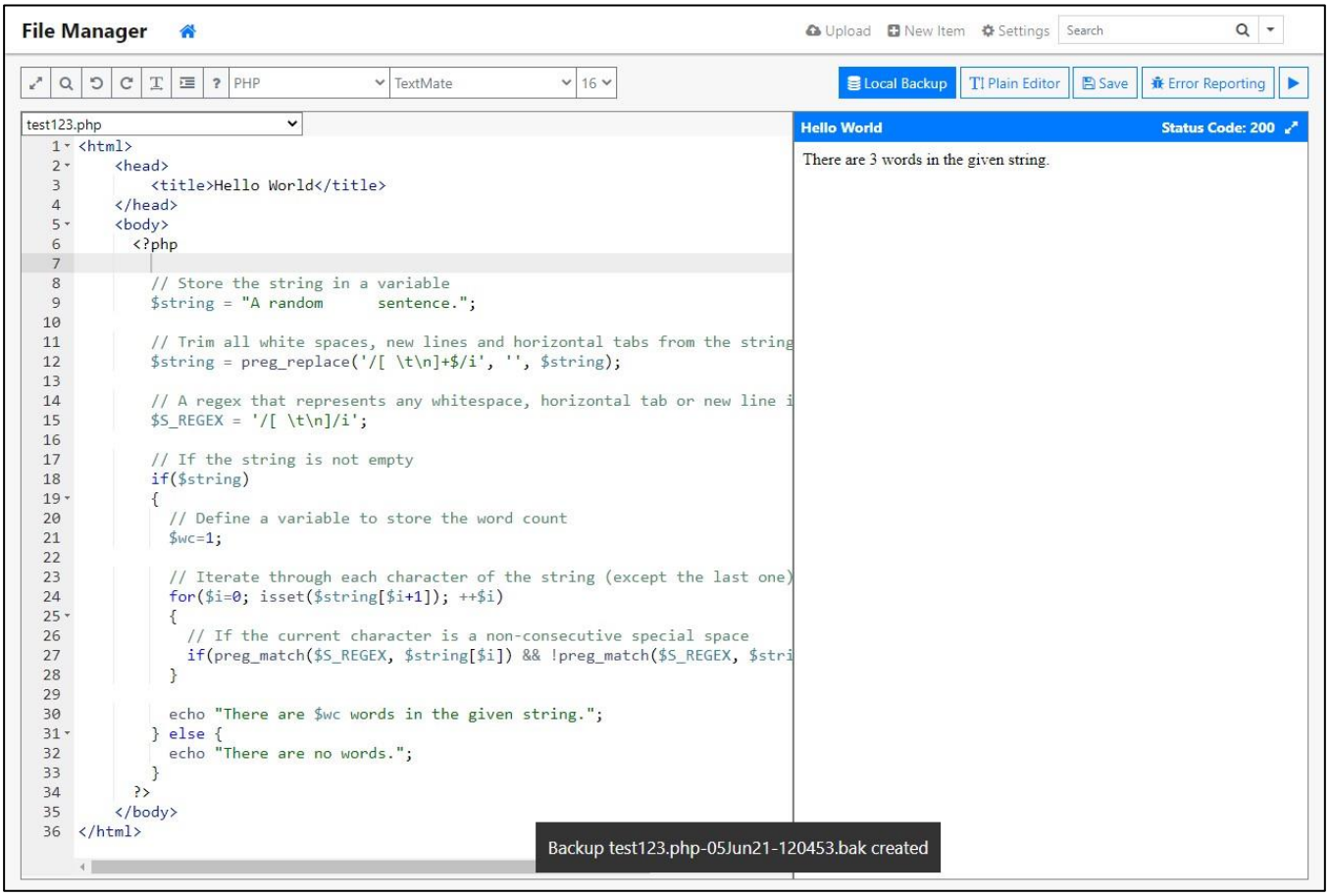


Fig. 6.3.27: Creating a backup for the file test123.php

14. Saving the file

A file can easily be saved either by clicking the Save button (that has been highlighted in the below image) or by pressing **Ctrl+S** when the editor is in focus.

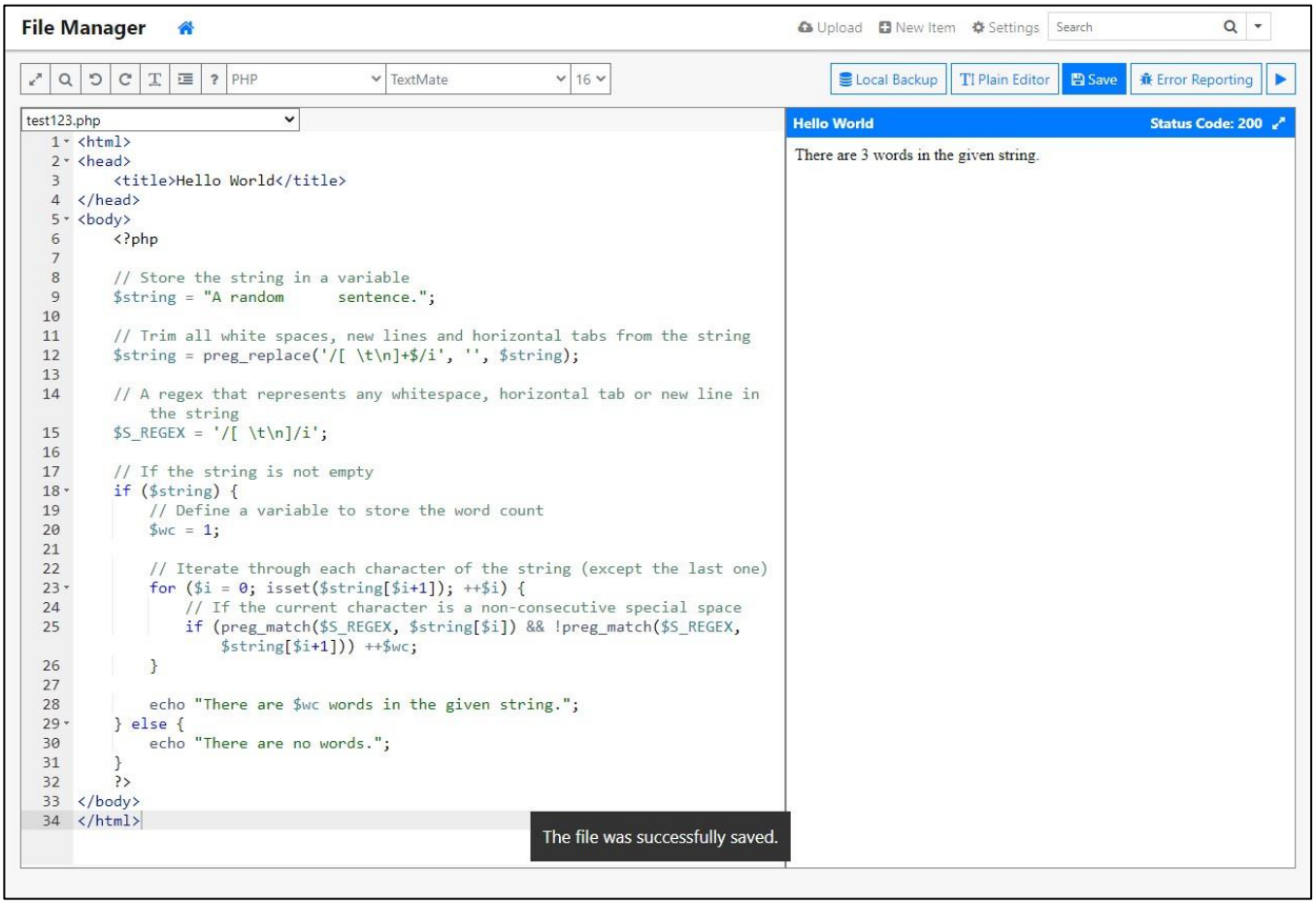


Fig. 6.3.27: Saving the current file

15. Hot Reloading/Live Reloading

This file editor supports the concept of hot reloading/live reloading. Basically, whenever we save the file via the editor by any means, the output screen would automatically refresh as soon as the file get successfully saved. This not only adds to the convenience of the programmer, but also enables the programmer to quickly develop/debug scripts/applications on-the-go without much hassles.

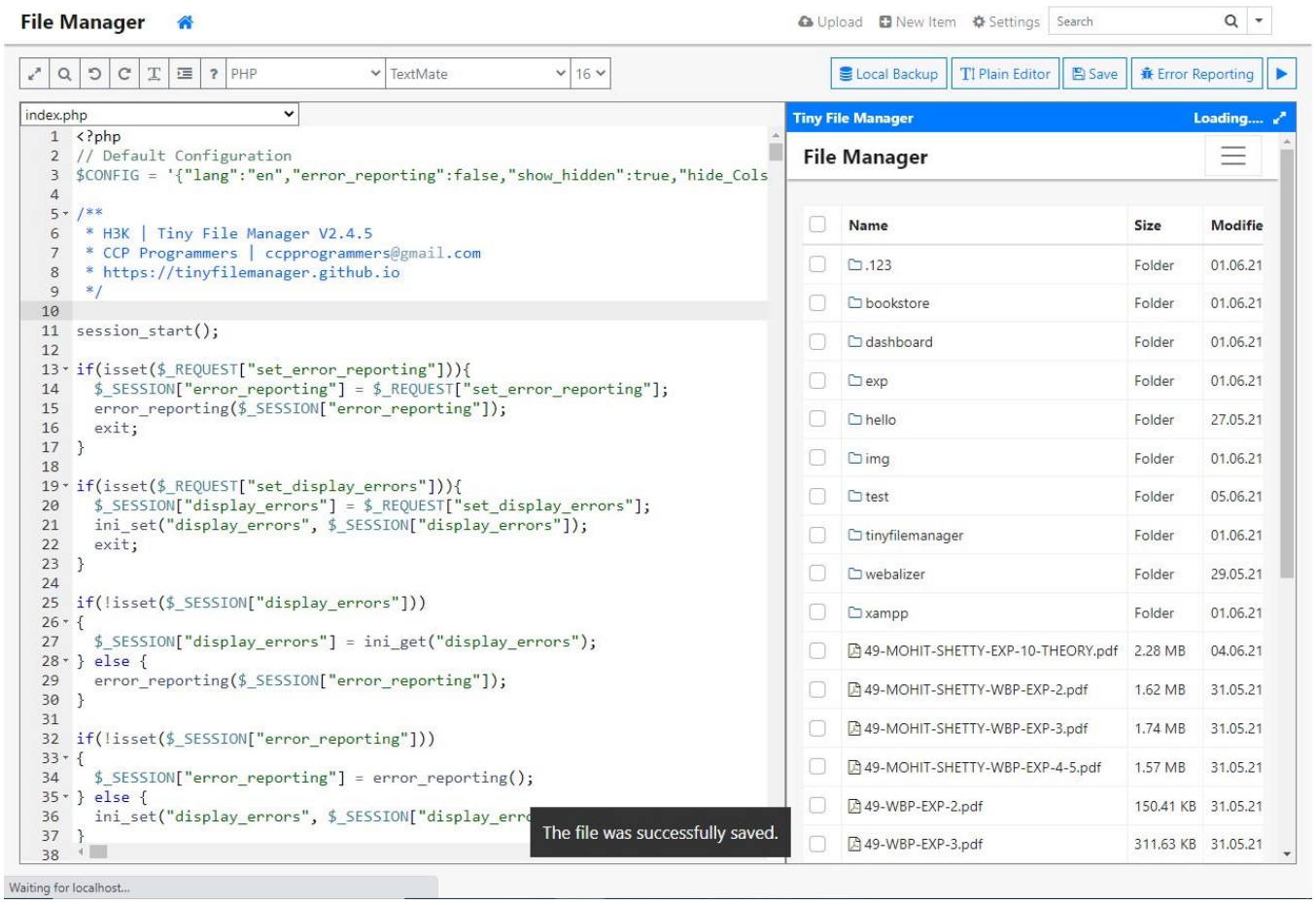


Fig. 6.3.28: A window that is currently auto-reloading the page (after saving the file)

16. Error Reporting

Our application even gives complete control over changing how errors shall be reported by the output window. One can decide which errors shall be reported (by default all errors are reported in debug mode [E_ALL]) or whether the reported errors shall be displayed or not.

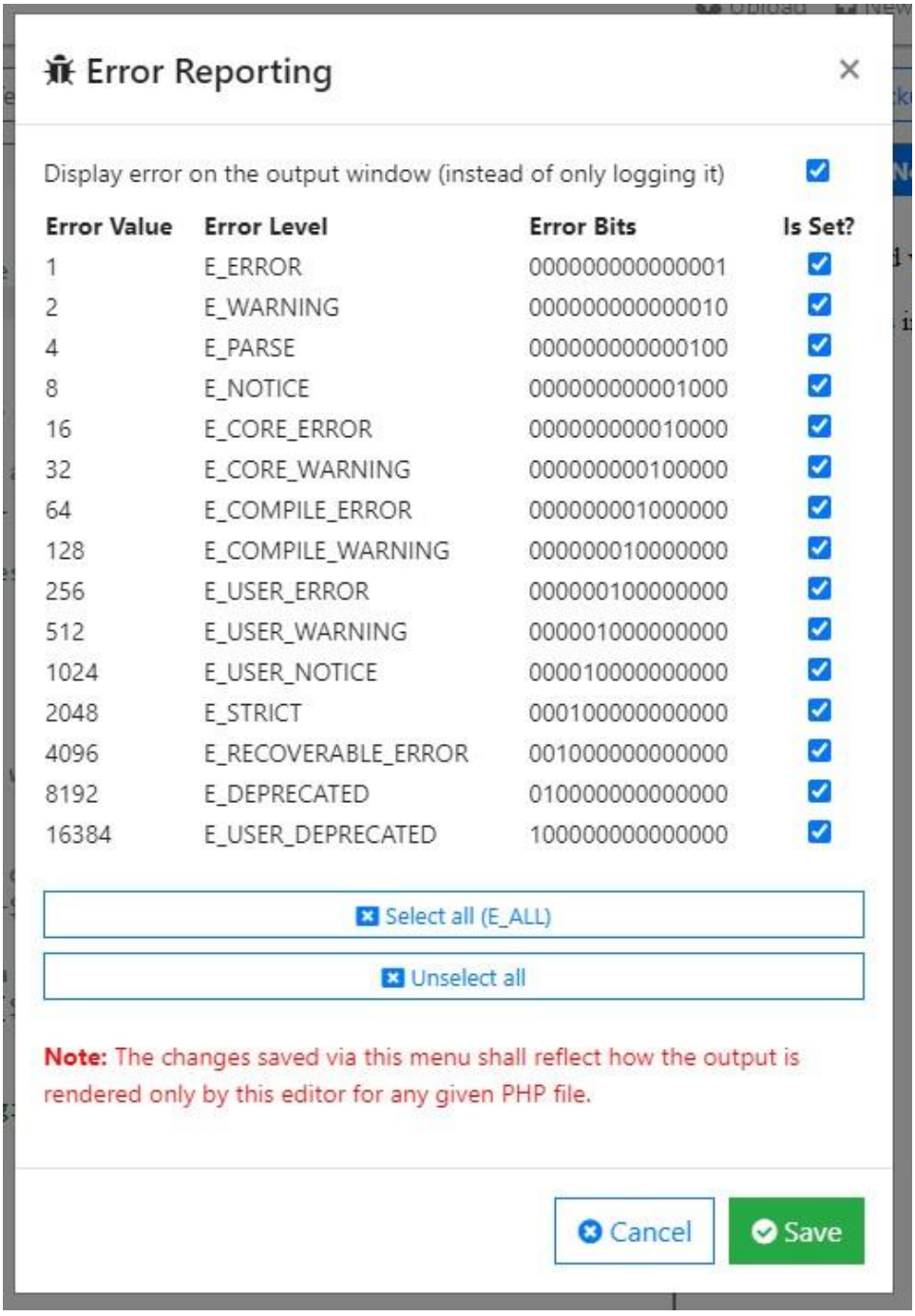


Fig. 6.3.29.1: Error Reporting Menu

To showcase the functioning of the error reporting menu, we will deliberately introduce an error by trying to assign an undefined variable ($a) to undefined variable ($a) which would raise a notice. We'll take a screenshot of that notice along with the code, disable E_NOTICE, save the settings and again check the output screen.

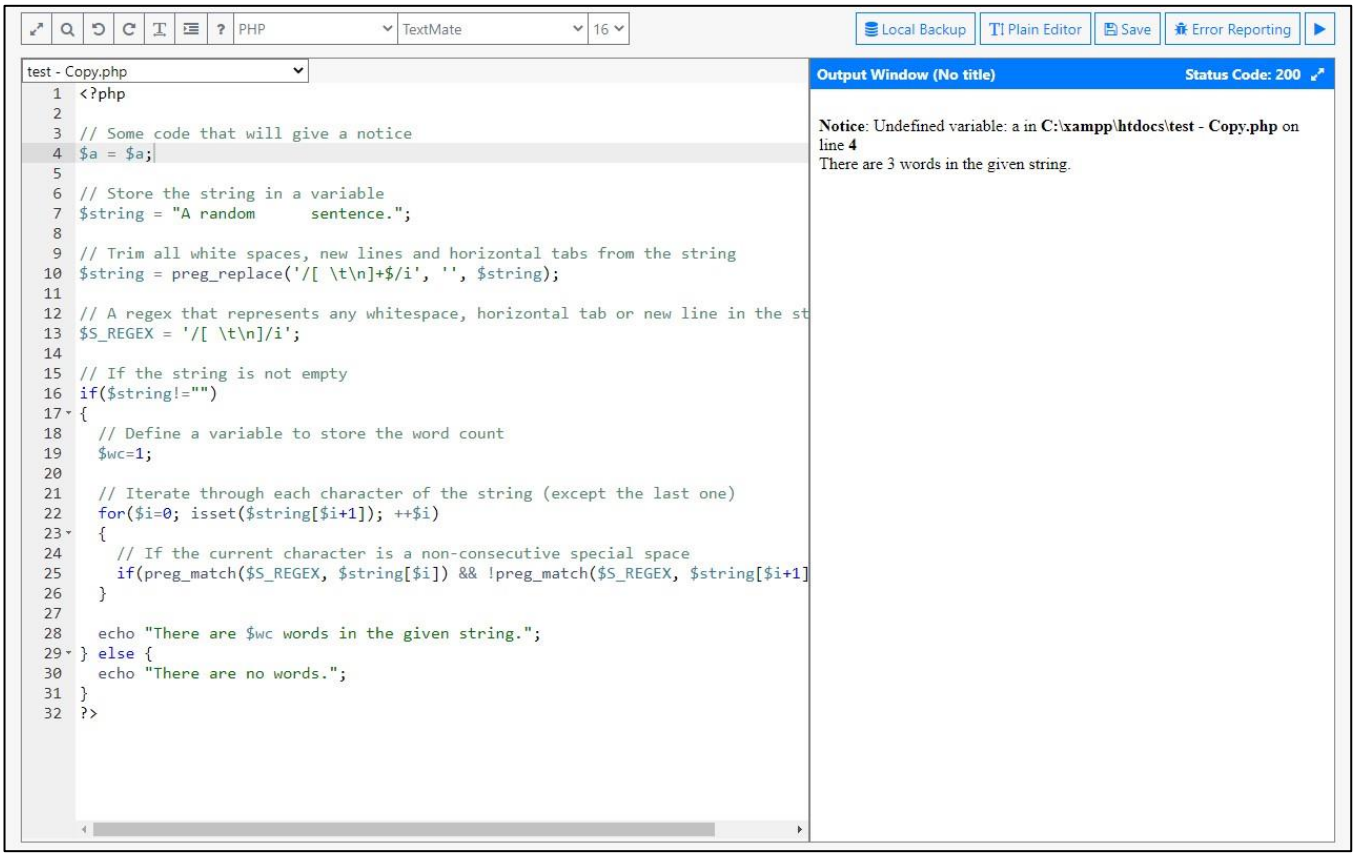


Fig. 6.3.29.2: Before disabling E_NOTICE

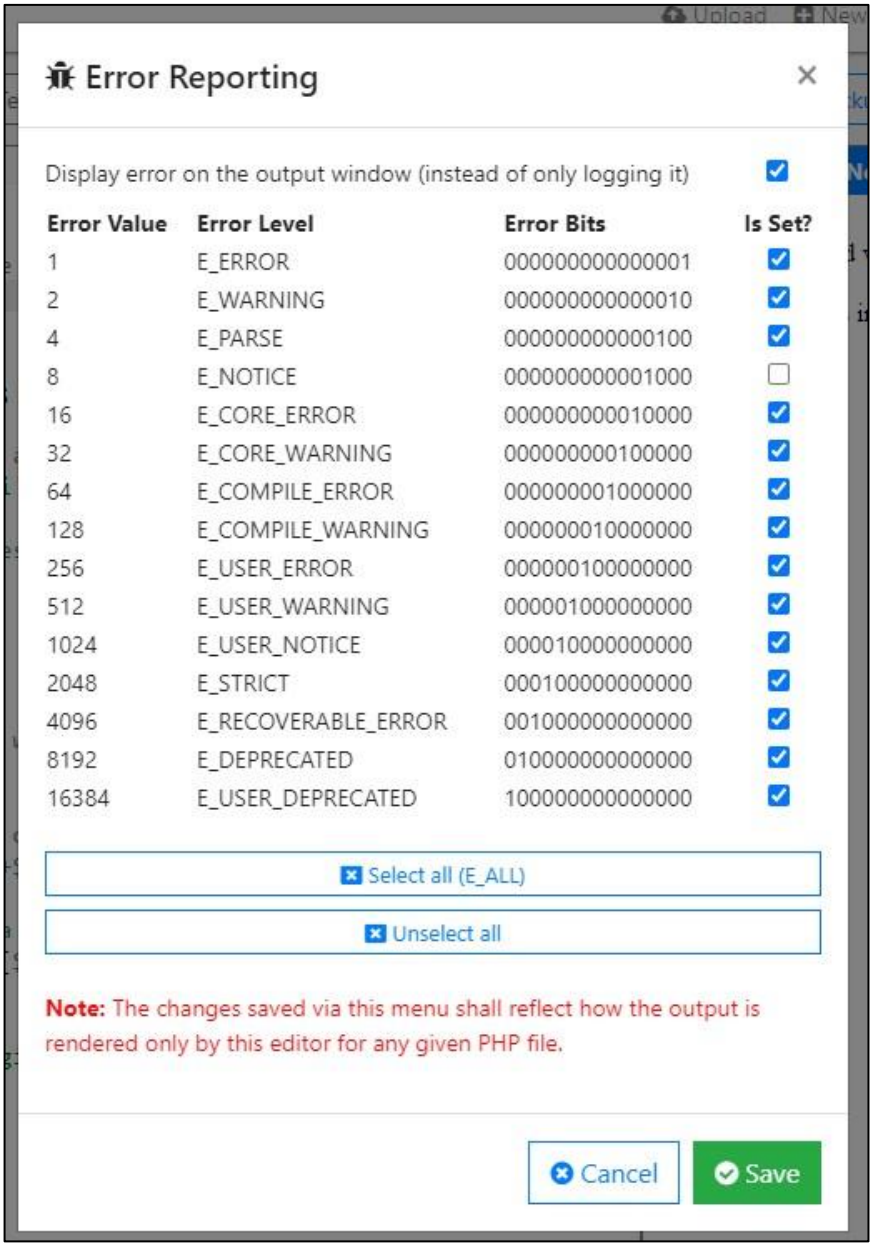


Fig. 6.3.29.3: Disabling E_NOTICE

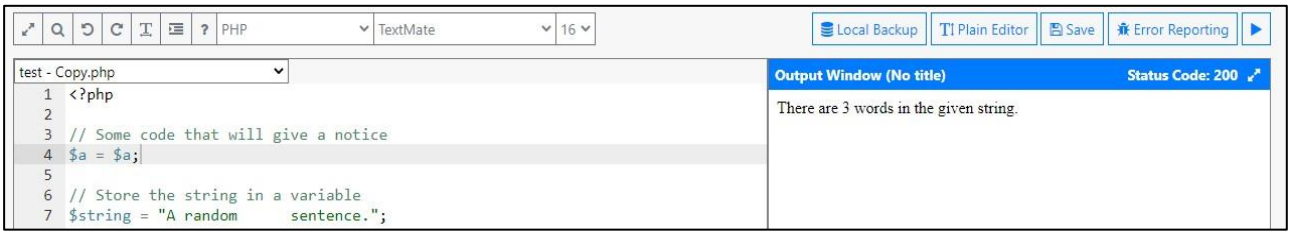


Fig. 6.3.29.4: After disabling E_NOTICE

CHAPTER 7

CONCLUSION

Core objective of the project was to learn how to use PHP and Mobile application development and JSON to make our modules and the report generator eases the work of the teachers and student and the classroom API also reduces the work of writing down roll no traditionally and using our IDE we can see results in real-time hence we made apps which reduce the load of teachers and student of MSBTE Using these modules the time and workload of teachers and students will be reduced and using PHP ide we can also implement better as we will already know how it going to look so having less error and as we have error sorting in php we can sort the errors according to their types the classroom API can also be used in every school or college situation to identify whose submission are left and to inform the students hence we fulfilled our main motive of making things easier by automating them and we learnt a lot about the mobile application development on how to make apps and in php we learnt how to use library and implementation of the library and we also learnt how to make realistic java Swing program/interface and we learnt how to use json which is used when fetching data from database which goes to the website we also learnt web scraping which is essentially a process to retrieve information from the internet and appending them in the program we also used Apache poi to make Excel sheets this library is used to create automation Excel sheets we also used ace.js and we learnt how to perform test cases, testing on programs and why they are performed we conducted numerous testing like white box testing black box testing and stress testing etc to make sure the program was working correctly hence giving us practical experience of how things work.

CHAPTER 8

REFERENCES AND BIBLIOGRAPHY

REFERENCES:

[1] www.google.com/en//selfdrivingcar/files/reports/report-0216.pdf

[2] https://www.dspace.com/zh/zho/home/products/systems/ecutest/configuration_examples/ecus_for_vehicle_dynamics.cfm

[3] http://www.mtc.umich.edu/test-facility

[4] https://en.wikipedia.org/wiki/Google_self-driving_car

[5] http://www.bitrebels.com/technology/teslas-first-autopilot-crash-happ ened/

[6] http://www.rand.org/content/dam/rand/pubs/research_reports/RR400/ RR443-2/RAND_RR443-2.pdf

[7] M. Aeberhard, S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, and N. Kaempchen, "Experience, Results and Lessons Learned from Automated Driving on Germany's Highways," IEEE Intelligent Transportation Systems Magazine, vol. 7, no. 1, pp. 4257, 2015.

[8] F. Flemisch, A. Schieben, N. Schoemig, M. Strauss, S. Lueke, and A. Heyden, "Design of human computer interfaces for highly automated vehicles in the eu-project HAVEit." pp. 270-279. 167 Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on May 06,2021 at 04:13:46 UTC from IEEE Xplore. Restrictions apply.

[9] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges," Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 368, no. 1928, pp. 4649-4672, 2010.

[10] O. Carsten, and L. Nilsson, "Safety assessment of driver assistance systems," European Journal of Transport and Infrastructure Research, vol. 1, no. 3, pp. 225-243, 2001.

[11] European research project "Testing and Evaluation Methods for ICT-based Safety Systems (eVALUE)", http://www.evalue-project.eu/

[12] M. Anwar Taie, "New trends in automotive software design for the challenges of active safety and autonomous vehicles."

[13] C. Gühmann, J. Riese, and K. von Rüden, "Simulation and Testing for Vehicle Technology."

[14] R. Schilling, and T. Schultz, "Validation of Automated Driving Functions," Simulation and Testing for Vehicle Technology, pp. 377-381: Springer, 2016.

[15] M. Hjort, H. Andersson, J. Jansson, S. Mårdh, and J. Sundström, "A test method for evaluating safety aspects of ESC equipped passenger cars: a prototype proposal," 2009.

[16] F.-Y. Wang, X. Wang, L. Li, P. Mirchandani, and Z. Wang, "Digital and construction of a digital vehicle proving ground." pp. 533-536.

[17] W. Huang, D. Wen, J. Geng, and N.-N. Zheng, “Task-specific performance evaluation of ugvs: Case studies at the IVFC,” IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pp. 1969-1979, 2014.

[18] H.-M. Huang, K. Pavek, J. Albus, and E. Messina, "Autonomy levels for unmanned systems (alfus) framework: An update." pp. 439-448.

[19] M. Broy, "Challenges in automotive software engineering." pp. 33-42.

[20] M. R. Heinen, F. S. Osório, F. J. Heinen, and C. Kelber, “SEVA3D: Autonomous Vehicles Parking Simulator in a three-dimensional environment,” INFOCOMP Journal of Computer Science, vol. 6, no. 2, pp. 63-70, 2007.

[21] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration." pp. 2641-2646.

[22] R. Austin, Unmanned aircraft systems: UAVs design, development and deployment: John Wiley & Sons, 2011. [23] J. L. Pereira, and R. J. Rossetti, "An integrated architecture for autonomous vehicles simulation." pp. 286-292. [24] https://www.tassinternational.com, TNO spinout TASSInternational

[25] D. Gruyer, S. Pechberti, and S. Glaser, "Development of full speed range ACC with SiVIC, a virtual platform for ADAS prototyping, test and evaluation." pp. 100-105.

[26] M. Lesemann, A. Zlocki, J. Dalmau, M. Vesco, M. Hjort, L. Isasi, H. Eriksson, J. Jacobson, L. Nordström, and D. Westhoff, “A test programme for active vehicle safety–detailed discussion of the eVALUE testing protocols for longitudinal and stability functionality,” in 22nd Enhanced Safety of Vehicles (ESV) Conf., Washington, USA, 2011.

[27] M. Buhren, and B. Yang, “Simulation of automotive radar target lists using a novel approach of object representation,” in 2006 IEEE Intelligent Vehicles Symposium, 2006, pp. 314-319.

[28] A. Sidhu, “Development of an Autonomous Test Driver and Strategies for Vehicle Dynamics Testing and Lateral Motion Control,” The Ohio State University, 2010.

A Fast and Reliable Real-Time Storage Interface Using File System Watcher

Mohit Shetty[1] Parth Mehta[2] Talha Shaikh[3] Maitry Dave[4] Ms. Vaishali Rane[5]
[1,2,3,4]Student [5]Guide
[1,2,3,4,5]Department of Computer Engineering
[1,2,3,4,5]Thakur Polytechnic, Kandivali, Mumbai, Maharashtra, India

Abstract— Storing, retrieving and manipulating data is one of the most common operation performed by on a normal computer system or a server machine. Data can be stored in different formats such as plain text, binary or custom formats (e.g. JSON, XML, HTML) and in different storages such as main memory (RAM), physical memory (a file system that interfaces a hard drive or any other physical storage) or on some cloud storage. The fastest way to access data on an average computer system would be to access it from its processors' registers or dedicated cache memory, but this might not always be possible and such storages generally have a limited capacity. The next fastest way to access data is the main memory of a system. The average time needed to access the memory can vary from 0.5ns (90% of the time) to 10ns (10% of the time) [1]. However, the data stored in the main memory is temporary and is lost as the power supply gets disconnected. The next option, which is present in most systems is to use a file system that would ensure that the data is permanently stored but the time needed to access it would be comparatively longer. In order to create a fast and reliable storage interface, we'll need to combine the positive points of both the main memory (fastest access time) and file storage system (reliability), with the help of a common file system watcher that keeps track of all the interfaces and this paper exactly discusses a concept based on this idea on an abstract level. The main memory shall be used to access the data and the file system to reliable store it. Whenever the data gets updated on the file system, the file system watcher shall notify all the relevant interfaces and update the main memory. The same notification(s) can then be used to make the storage interface real-time.

Keywords: Storage Interface, Real-Time, Fast, Reliable, File System Watcher, Storage Model, Data Retrieval, Data Storage

I. INTRODUCTION

A. What is a storage interface?

A storage interface can be anything that interfaces a given set of storage(s) or storage interfaces(s) or storage system(s) in order to logically provide a convenient way to provide programmers an abstracted and convenient way to store, manipulate and access data. The storage interface can then be used while either developing programs or a server that could require this concept to create custom storage designs. At minimum, a storage interface should define a way to store and retrieve data. The storage interface described in this paper interfaces mainly two types of storages – the file system and the main memory.

B. What does this paper mean by the terms fast, reliable and real-time (storage interface)?

A fast storage interface is a one that can return the data stored in the memory, in the shortest time possible. A reliable storage interface is a storage interface that makes sure that the data that is once successfully stored, doesn't get lost in case of sudden power loss or any other circumstance and restarting the system would resume things for all the connected reliable components as normal. A real-time storage interface is an interface that is capable of notifying its listeners in case of any change in data.

C. How does the File System Watcher API help?

The File System Watcher API helps by notifying the custom file system watcher of syncing the file system with the main memory and making the interface real-time is played by the File System Watcher API. Whenever there is a change in the file that is being interfaced by an interface, the file system watcher will notify (all the) interface(s) that interface that file in response to which the interface would first update the main memory that it has interfaced and then re-send the notification to all its listeners (that the data has been updated), either locally or via network, which could then either retrieve the newly updated data or simply just ignore the request based on its own state.

II. DESIGN OVERVIEW AND FUNCTIONING

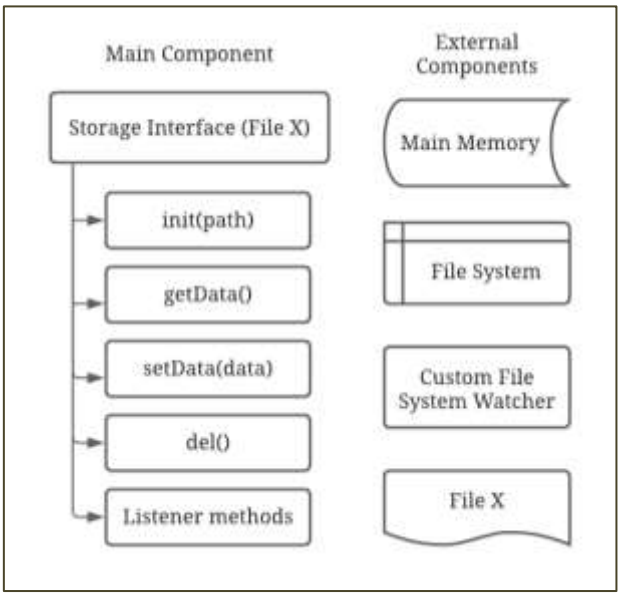


Fig. 1: Design Overview

The main component as mentioned in the title of the paper is the Storage Interface. Each storage interface interfaces a file whose path is provided to it. By default, any storage interface would have four methods defined in it–

1) init(path):

The init method shall be called to initialize the storage interface. The path passed to this method, is that of the file to be interfaced. The method gets the data stored in the interfaced file into the memory and registers itself to the custom file system watcher to get notified about the changes. The method is called implicitly while creating the object and the user. If the file isn't present, an exception shall be thrown/raised for the caller or the object creator to handle. The path given to this method should be stored by the interface for future use (e.g. the setData method).

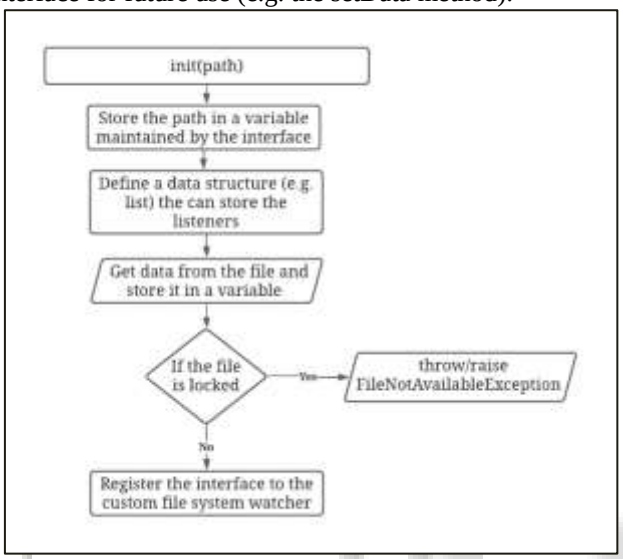


Fig. 2: An example flow that initializes a storage interface

2) getData():

The getData method returns the data stored in the main memory (in high level programming languages as a private variable). It by default does not accept any parameter, however named/optional arguments can be added later while extending this concept further.

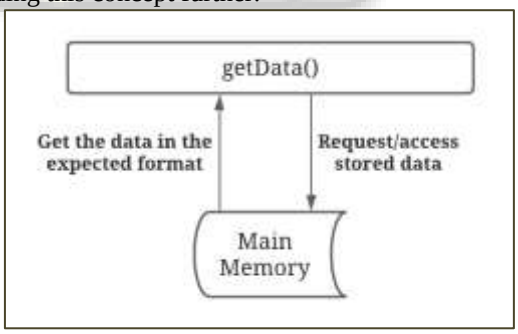


Fig. 3: A diagram that represents the functioning of the getData method

3) setData(data):

The setData method by default accepts a single parameter, where the first parameter represents the data to be updated in the file (and then implicitly the main memory). The setData method tries to write to the interfaced file. (Note: The file isn't permanently locked by the storage interface) If it fails to write either because the file is locked or because the file isn't available (moved/deleted) at the path given while implicitly calling the method, then two unique exceptions shall be raised/thrown for each case. Once the file is successfully updated/modified, a custom file system watcher which shall implicitly update the main memory (the private variable). How the custom file system watcher updates the main memory has been described in detail, while discussing the external components that the interface uses.

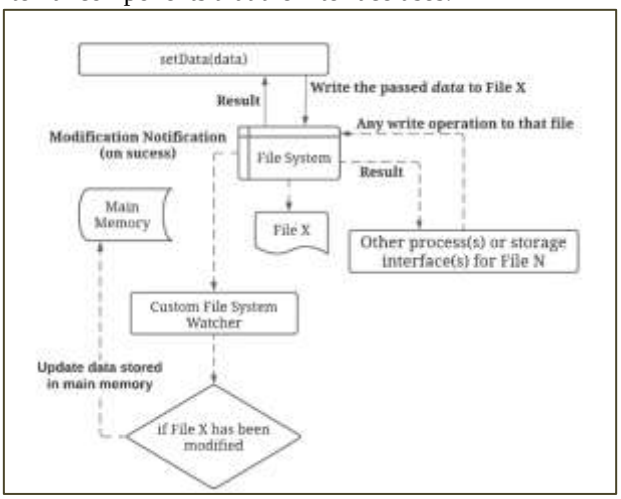


Fig. 4: A diagram that shows how the setData method should work with other components

4) del():

The del method is called when the storage listener is about to get destroyed, either by the programming environment when it's reference is no longer accessible or because the programmer decides to explicitly delete it with the help of the options provided by the language. When this method is called, the main memory used to store the data and the path of the interfaced file is de-allocated and a request to unregister is sent to the custom file system watcher, after which the custom file system watcher would no longer check for the path of that interface (whenever it receives a notification) and the deallocated memory, would no longer be updated or visible for access. This method shall only be called once in the lifetime of a storage interface. Extra option/named parameters shall preferably not be given to this method while later extending the concept to avoid un-necessary complexity.

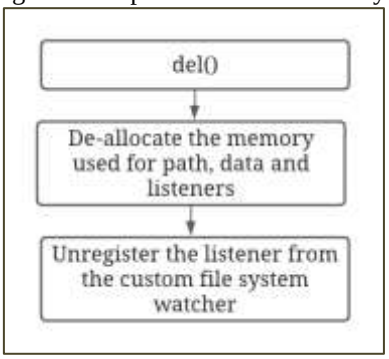


Fig. 5: Flowchart for destructor del()

Apart from the storage interface, there are 5 other external components that the storage interface uses, namely,

1) Main memory:

The main memory is the memory from which the storage interface would always access data from. Based on the language or level at which the storage interface is implemented, the main memory could be a (private) (pointer) variable defined (in a class or interface or namespace) inside a program, from where the data can be obtained via a function/method of the (abstract concept of) interface. This

component is indirectly accessed by the caller when the getData() method called.

2) File System:

The file system is a component provided by almost every operating system. It acts as an abstract for every form of physical storage it supports, and provides different primitives for managing information and primitives that can store those primitives (recursion). Most file system have the concept of files and directories to do most of the work and can have their own ways of addressing a given file, where (the drive/root directory and) every directory until the given file is separated by a forward slash (/) or back slash (\). To support a wider range of operating systems, one can use relative paths separated by forward slashes which would add to the portability of the system that is being designed. The main use of the file system here is to get data whenever the caller, calls this function and set the data, when the program explicitly asks to do so and to notify the File System API on the same.

3) Custom File System Watcher:

The custom file system watcher is a component that uses the File System Watcher API provided by the operating system to get notified whenever any file event takes place within a directory. The custom file system watcher would then iterate through the set of registered interfaces and see if any interface has a path matching to the modified file and update the data in those interfaces and notify the interface to notify its listeners (if any).

4) File X:

File X is the file that is being interfaced by the storage interface. The path of the file is stored by the interface when the init method is called. Whenever the data in this file is modified, the file system watcher API (OS level) gets called, which notifies the Custom File System to call the concerned listener interface methods.

5) Listener methods:

Listeners in this context refer to functions that get called whenever the data of an interface gets updated in any way. In abstract language, they listen to the interface for data changes. A listener can listen to create/modify/delete changes based its type. The custom file system watcher calls the onCreate/onModify/onDelete which in turn selectively calls the listeners attached to that interface.

III. Steps to Implement the Interface

1) Read the paper to get a rough understanding of the idea before you actually start implementing this concept.
2) Decide a common unique directory to specifically store data files for the system you are developing (which would help in avoiding overhead events from unexpected).
3) Create a file for the program/script.
4) Import the required libraries (e.g. watchdog,)
5) Design the custom file system event handler via a class/interface/namespace as defined in the design overview section.
6) Write a few classes to represent every listener uniquely based on its type (helps in using a single queue to store the listeners)
7) Design the main storage interface class as defined in the design overview section.
8) Use the storage interface concept in your program/server (or just simply use an infinite loop while making some minor modifications).
9) Test the code. (To generate the below output we, created a text file called sample.txt, saved it once, deleted it and then saved the main script file the was being executed as a process.)

IV. Extending the Concept Further

This concept in itself can be extended further by supporting other data types or formats of data like JSON or XML and by mapping it to the types offered by the programming language or defined by the user. Alternatively, one could provide better means to conveniently update/manipulate the data by providing new methods that can duplicate the data in the main memory, modify it and update the interfaced file with it or modify the existing setData method's parameters to accept different parameters based on the

V. Making the Interface Real-Time

A. Within a program/process

By default, the interface would be real-time in itself as the interface would accessible from within the program, so the user can directly call getData to fetch the latest data without relying on any listener. However, if the program is GUI based, then one could attach a listener to listen to create/modify/delete events and filter out the events and update the UI with the latest data/state in the function itself. This would conveniently help to ensure that the user is always viewing the latest data.

B. Across different programs/processes (locally)

It is always better to link the programs of the system before executing them in such a way that the programming itself safely manages the data/object sharing part (by implementing a common file system watcher for all the processes). However, in case if that isn't available, then the programmer could directly use file system as the common point to get notified about all the changes while maintaining separate memory for each process (basically creating an interface as and when needed as per the state and requirements of the process) or by using the concept of multiprocessing which may or may not be supported by the language or use a design similar to the network part.

C. Over the network (as a deployed server)

A server that has already been created can use this model to efficiently store and retrieve data. If the client wants to get notified about data changes in real time, then the server can support the (secure) web socket protocol (ws(s)). Whenever the client requests a connection for a specific interface, the server can attach a listener to the interface and notify the client about the changes via the connection. Now it would be, entirely up to the client (application) whether or not it wants to fetch for new data or not. Making the state of the connection synchronous to the state of the storage interface, or maintaining a common unique pool for the storage interface (that can be uniquely identified by their path).

VI. ADVANTAGES

1) The storage interface concept, in its simplest form can be claimed to be the fastest for access for a given computer system as programmatically accessing the memory does not have any visible overhead (except if any by the programming language).
2) Implementing the system from the perspective of a programmer in its simplest in any language where the file system watcher does not have an unfamiliar model is very easy once the core concept of the entire interface is understood well.
3) The reliability provided by the interface is never at the cost of the access time or the guarantee of successfully getting the data as the memory allocated for every new update is separate, until the reference/starting memory location gets assigned to the (pointer) variable. So even if the data of the interface is accessed while the data is getting updated, the current data would be returned until the new data isn't available.
4) The cost and resources needed for the implementation of this system is very low as anyone with basic but accurate knowledge of computer science and how operating systems can implement this system and the chances of malfunctioning are very low, unless the concept is extended or enhanced for any reason.
5) This concept can be used to build better and faster servers, assuming that other factors such as security and hardware reliability are taken into consideration.

VII. PROBLEMS AND COUNTER MEASURES

1) While designing a system with this concept, a lot of things would be needed to taken into consideration such as the constraints of the hardware, number of active users, how often the data updates, in order to avoid unexpected behaviour once the system is deployed. Reading this paper carefully and performing enough research before implementing big systems with this system can help counter these problems. Creating local real-time programs with the concept of interfaces without prior research shouldn't be a problem.
2) The reliability of the system might not be sufficient for all use cases, in which case parallel or a different non-volatile system(s) can be used to store data assuming it has an interface that can be listened to for changes or an interface can be programmed over the existing interface to listen to create/modification/delete based on the local/network device and the storage model being used.
3) Not all file systems might use the same method/primitives to address a given file, so to avoid problems while porting the system from one computer to another with different OS, relative addressing, where each directory reference would be preferably separated with forward slashes. Alternatively, the methods provided by the libraries of a programming being used can be used to join directories and file to form a path or to resolve a relative path into absolute can be used to solve such problems for most of the operating systems that are commonly used.
4) Having too many storage interfaces registered with the custom file system watcher could be end up causing performance issues as the watcher would have to iterate through all the interface would have to iterate through every interface in order to notify the right interfaces, and the interface itself would have to iterate through all its different kinds of listeners that could make things a lot slower. In order to solve this as an issue, a common unique pool of interfaces can be created, where every interface would be uniquely known by the path it interfaces. This would eliminate redundant iteration done by the custom file system watcher. At the level of the interface, for every kind of listener a different queue can be allocated which could reduce the number of iterations but would need more memory for every interface.
5) The idea by default is very plain in itself and hence custom mechanism to retrieve, manipulate or update the data might not be clear in the mind of the programmer. Researching on the format that needs to be implemented and mapping the format to an existing library class or user-defined class might need to by the programmer.

VIII. APPLICATIONS

1) It can be used in local programs to create objects from which we can logically, directly read and write data to.
2) It can be used in GUI-based applications to design dynamic UIs that update when there is a change in data (or the contents of a file).
3) It can be used in servers that quickly need to serve dynamic data as soon as possible.
4) It can be used in applications that need to serve data in real-time to all its clients without any (major) delay or manual request by the user to check for the validity of the data.
5) It can be used to cache commonly accessed files dynamically with the help of a fixed length priority based queue consisting of storage interfaces, interfacing those files.

IX. CONCLUSION

The storage interface described in this paper allows a programmer/system designer to reliably store and quickly access that data in the fastest possible way for a given operating system. By using this interface model, one can conveniently store data for simple programs or GUI-based applications that display dynamic data or different types of server. By writing this paper, we not only described and brainstormed our idea but also related it with its possible real world application.

REFERENCES

[1] The random memory access time data was taken from the highlighted points of the following article: https://www.cs.uaf.edu/2011/spring/cs641/lecture/04_05_modeling.html#:~:text=Average%20Memory%20Access%20Time%20(AMAT)&text=For%20example%2C%20if%20a%20hit,1.4ns%20average%20access%20time.
[2] M. Mesnier, G. R. Ganger and E. Riedel, "Object-based storage," in IEEE Communications Magazine, vol. 41,

no. 8, pp. 84-90, Aug. 2003, doi: 10.1109/MCOM.2003.1222722.

[3] Y. Kang, Jingpei Yang and E. L. Miller, "Object-based SCM: An efficient interface for Storage Class Memories," 2011 IEEE 27th Symposium on Mass Storage Systems and Technologies (MSST), 2011, pp. 1-12, doi: 10.1109/MSST.2011.5937219.

[4] Mary Baker, Satoshi Asami, Etienne Deprit, John Ouseterhout, and Margo Seltzer. 1992. Non-volatile memory for fast, reliable file systems. SIGPLAN Not. 27, 9 (Sept. 1992), 10–22. DOI:https://doi.org/10.1145/143371.143380

[5] A. K. Paul, R. Chard, K. Chard, S. Tuecke, A. R. Butt and I. Foster, "FSMonitor: Scalable File System Monitoring for Arbitrary Storage Systems," 2019 IEEE International Conference on Cluster Computing (CLUSTER), 2019, pp. 1-11, doi: 10.1109/CLUSTER.2019.8891045.

[6] D. Peric, T. Bocek, F. V. Hecht, D. Hausheer and B. Stiller, "The Design and Evaluation of a Distributed Reliable File System," 2009 International Conference on Parallel and Distributed Computing, Applications and Technologies, 2009, pp. 348-353, doi: 10.1109/PDCAT.2009.37.

[7] Gackenheimer C. (2013) Using the File System. In: Node.js Recipes. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4302-6059-2_3

[8] H. El-Rewini, H. H. Ali and T. Lewis, "Task scheduling in multiprocessing systems," in Computer, vol. 28, no. 12, pp. 27-37, Dec. 1995, doi: 10.1109/2.476197.

[9] H. El-Rewini, H. H. Ali and T. Lewis, "Task scheduling in multiprocessing systems," in Computer, vol. 28, no. 12, pp. 27-37, Dec. 1995, doi: 10.1109/2.476197.

[10] Qveflander, N. (2010). Pushing real time data usingHTML5 Web Sockets (Dissertation). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:umu:diva-36530

[11] Z. Deng and J. W. -. Liu, "Scheduling real-time applications in an open environment," Proceedings Real-Time Systems Symposium, 1997, pp. 308-319, doi: 10.1109/REAL.1997.641292.

[12] SYSTOR '15: Proceedings of the 8th ACM International Systems and Storage ConferenceMay 2015 Article No.: 8 Pages 1 6 https://doi.org/10.1145/2757667.2757670

[13] Rhea, Sean C., Russ Cox, and Alex Pesterev. "Fast, Inexpensive Content-Addressed Storage in Foundation." In USENIX Annual Technical Conference, pp. 143-156. 2008.

