



Yohannes Loga

# Client Side Collaboration of eLearning Platform and Integrated Development Environment

Helsinki Metropolia University of Applied Sciences  
Bsc

Media Engineering

Thesis

Date 18 May, 2011

Author(s) Title Number of Pages Date	Yohannes Loga Client side collaboration of eLearning platform and integrated development environment 64 pages 13 April 2011
Degree	Bachelor of Engineering
Degree Programme	Media Engineering
Specialisation option	Application development, Mobile programming
Instructor(s)	Mika Lackman, Project Manager Kari Aaltonen, Principal Lecturer
<p>The project goal was to provide the PHP software programming course materials, for a company called Viope Solutions Oy and to develop an Eclipse plug-in for the collaboration of Eclipse editor and their server. In this paper, the concept of E-learning system is briefly discussed, and the company's current system is also demonstrated along with the PHP course material that was developed for it. Demonstrating a plug-in that was developed for facilitating this system is also the other purpose.</p> <p>The development of plug-in was proved to be of great importance, because it helps users of this system to use the widely used Eclipse editor to perform programming exercises. The application, therefore, should improve the popularity of the company and increase their efficiency. The proposed solution is based on our experience as students on the editor and on the fact that it is a well known one.</p> <p>By the help of this project, it was proved that it is possible to integrate the Eclipse editor with the current system by developing a plug-in for it, which in turn yields a great result.</p>	
Keywords	plug-in, Eclipse editor

## Table of Contents

1	Introduction	1
2	Overview	2
2.1	Evolution	2
2.2	Types of E-learning	3
2.3	Pros and cons	4
3	System	6
3.1	Communication processes	6
3.2	Preferred technologies by learners	7
3.3	Virtual school	8
3.4	How do virtual school systems work?	9
4	Viope eLearning platform	11
4.1	Structure	11
4.2	Logging into the service & system panel	12
4.3	Learning material management & exam management	15
4.4	Extra material management & exercise bank	20
4.5	Evaluate exercises & student followup	23
4.6	Communication & help	24
4.7	Chapters	25
5	Eclipse plugin for Viope system	28
5.1	Accessing the system	28
5.2	Data translation and transformation	33
5.3	SAX parser	35
5.4	DOM parser	37
5.5	Accessing chapters and exercises	37
5.6	Sending programming exercises	41
6	How the plug-in works	42
6.1	The first proposal for the application	42
6.2	The application done	47
7	Conclusion	50
	References	51

## 1 Introduction

The goal of the thesis is to utilize the Integrated Development Environment of Eclipse (IDE) in order to increase the quality of learning sessions in Viope's platform. The usage of IDE in learning software development should give a new perspective for the learning process. The features supported by Eclipse as professional Integrated Development Environment give more detailed information and more versatile tools to help the learner to get a deeper understanding. The main tasks are to develop PHP course materials and exercises and the mechanism of exchanging tasks through Eclipse IDE. The whole project contains two aspects, the client side collaboration eLearning platform and integrated development environment and the server side eLearning platform and integrated development environment, the former of which will be addressed in this paper.

Internet has grown fast for the last couple of years, affecting the lives of each of us. It is a fact that the growth has speeded up the learning process. When internet and learning system are mentioned the word E-learning pops to head. In this paper the general concept of E-learning system will be discussed, along with the E-learning material that was done for a company called viope. The learning materials provided are for the programming language called PHP.

In order to give a clear idea of the E-learning system viope has, the parts of the system are discussed thoroughly and demonstrated. Also the reason for developing a plug-in for this company will be analyzed. It must be noted here that the purpose of this paper is not to demonstrate how to build a plug-in, instead to show what kind of a plug-in it is and how it works.

## 2 Overview

### 2.1 Evolution

Sitting down in the class rooms and following lectures of teachers had been the only way to learn, until about 25 years ago when the multimedia era started to evolve. The technological tools like CD ROMS and power point became powerful tools in delivering education to learners. Despite the big advantages these technologies brought to the society, they lacked sufficient interaction between the instructor and the learners which in turn slowed down the learning process and made it somewhat inefficient.

Roughly between the years 1994-1999 the web was starting to bloom, which contributed a great deal to the world of e-learning. At that time email, web browsers, HTML and media players were used by trainers to deliver a more interactive and better way of learning. In the twentieth century better technological tools started to emerge, which made possible the merging of live instructor led training with the web. To mention some of these technologies Java/IP network applications, rich streaming media, high-bandwidth access and advance web site design are some of them. [1]

Before we go any further lets define what E-learning really is. Electronic learning commonly known as E-learning is a means of training by incorporating self-motivation, communication, efficiency and technology. Since there is no need to be in the same room with the trainer, E-learning avoids distance problems. But it will be up to the students to deliver the tasks given by the trainer which might not be close by so they need self-motivation to do so.

The structure of E-learning can be divided into self-paced, instructor led, and self-study with expert systems. The self-paced structure needs more self-motivation from the student because learning materials are provided to the student and is expected to complete the training. While in the case of instructor led, the student is guided by an instructor to achieve the tasks. And in self-study with an expert the students are monitored by a trainer but still expected to take the responsibility of finishing the tasks by themselves, so this structure incorporates both the self-paced and instructor led systems.[2,3]

## 2.2 Types of E-learning

E-learning can be achieved by real time communication using technologies such as video conferencing, teleconferencing and online chat programs, or it can be achieved by email, threaded discussions, blogs, and online forums. Hence E-learning types can be referenced as online and offline learning system.

Especially these days it is a common thing to “google” for blogs and threaded discussions in search of answers to our daily problems. From blogs a learner finds answers quicker and gets more solutions from different people than a trainer in a class room. As we saw at the beginning of the overview of E-learning system, there are different types of learning systems:

- Purely online – no face to face meetings
- Blended learning combination of online and face to face
- Synchronous, meaning the e-learning method used is real-time
- Asynchronous, meaning the e-learning method used is internet-based, network-based or storage disk-based modules
- instructor-led group
- self-study
- self-study with subject matter expert
- web based
- computer based (CD-ROM)
- video/audio tape [4]

While these are the detailed types of E-learning, some categorize them in to groups which they refer to as 3 tier system as demonstrated in table 1.

Table 1 Types of E-learning system [5]

Tier 1	Tier 2	Tier 3
low interactivity mainly text, multimedia or graphic one way communication	moderate to high interactivity has some degree of a learner to computer interaction	high interactivity includes learner to learner and learner to trainer interaction
Powerpoint presentation, learning on a personal digital assistant, e-books, podcasting, videotape, audiotape	Interactive resources, quizzes, tests, reflective learning, games, simulations, demonstrations	Virtual classrooms, streaming media, group games, videoconferences, audio conferences, chat groups, emails, discussion lists, blogging, wikis, moblogging.

### 2.3 Pros and cons

When designing a system, it is an important issue to take into consideration the pros and cons of the system as described in table 2. Even though it has a lot of advantages, one must consider the disadvantages and try to design in order to suppress them to develop an effective system.

Table 2 Pros and Cons of E-learning system [6]

Pros	Cons
no need to travel long distances to get the education, thereby reducing travel cost and time	lack of face to face interaction, which might be important in some cases
self-paced study system	difficult to distinguish the accredited E-learning system providers from the non-accredited ones
the freedom for the student to adjust the time according to his or her schedule	lack of motivation can lead to dropping out of the class
the possibility of choosing learning materials by a student, which he or she finds appropriate for his or her level	learning the system might be time consuming
the availability of chat rooms and discussion bulletin helps the student to contact other students and teachers at any time and get information	lack of social interaction can cause student to be isolated
computer skill development	unavailability of a teacher at a needed time and a slow internet connection can be frustrating to a student

By taking in to consideration the pros and cons given in the table 2, it is possible for whoever is building an E-learning system to develop the system effectively by trying to overcome the disadvantages of the system. [6]

### 3 System

#### 3.1 Communication processes

The communication process comprises of the people involved in the E-learning process and the technologies used to deliver it. When we talk about the people involved we talk about the creators and consumers of this system.

Designers, authors, writers, illustrators, photographers, animators, videographers and the like can be grouped under creators. While under consumers we have students or they can also be referred to as readers or users. In between the creator and the consumer there has to be a means by which education can be delivered with, this is done by using a host. A host makes it available over a network in order for students to access and help the trainers to provide the learning materials, maintain and administer it.

It is then time to view the technologies utilized by the creator, host and consumer in brief as shown in table 3. The technologies used are hardware, a network connection and software.

Table 3 Technologies used by creator [7]

	Creator	Host	Learner
Hardware	multimedia work- tation	Network server	Personal compu- ter
Connection	Moderate speed	High speed	Moderate speed
Software	Authoring softwa- re	Web server soft- ware	Browser and me- dia players

### 3.2 Preferred technologies by learners

When planning to develop a system, it is imperative to take in to account what are the preferred technologies used by a learner. By doing this it would be possible to develop a well formed and efficient one.

The specification of the computer used by students determines the design. For example in order to deliver a media file, the computer system of the learner should be able to support it.

From the specifications we can mention some like the processor type, how much memory it has, display size, colors in bit, hard disk size, CD or DVD type and audio type. If a survey is done it is possible to find out what is on the market and what really students have, thereby making it easy to design learning products.

When we go deeper in to the computer specifications, the speed of the computer depends on the processor type. The latest and fastest processor type is demanded, in order to play animations with a lot of sound and a good video quality. In case of memory it determines the multitasking property of the computer. Multitasking is the ability of accommodating many programs and manipulation of many data at once.

When considering the display, its size should be taken in to account. Because it determines what the learner sees, and the color depth determines the number of colors displayed at once. These specifications help when designing pages sizes, in order to know how many tasks fit in one window. It is a common experience that students get frustrated whenever they have to switch between different windows to perform one task.

In general it is not such a good idea to make software which can take up a lot of space, but it is a good practice to know what the learner computer accommodates according to the size of the hard disk they have. In case they have to download the software in their computer it is advised to know beforehand the size of hard disk available.

To deliver education which is electronically available, it is not necessary to have a network connection; instead it can be delivered by using CDs or DVDs. So it is a good practice to know what the students would use to take the lesson, or for what purpose they need the CD or DVD. It can just be for offering multimedia components or it can be the whole content. [7]

### 3.3 Virtual school

The offering of courses entirely or primarily through online methods by institutes is what is common referred to as a virtual school. This system includes tools for course authors, administrators, instructors, and learners. The availability of features needed to assemble, administer, and conduct courses as a complete package, makes virtual school system more efficient and consistent.

Instructors make use of this system, to build well-structured and integrated lessons that can be offered to students. This means that by using this system it is possible to post assignments, assign grades, route messages among participants, and conduct online discussions with the class. There by making it easier to store recording of test scores, tracking activities and reporting results.

And on students' side, they find it easy to get materials posted by the teacher, post assignments they have done and interact with their instructor and classmates. As an example we can see the web portal of Metropolia University as shown in the figure 1 below.

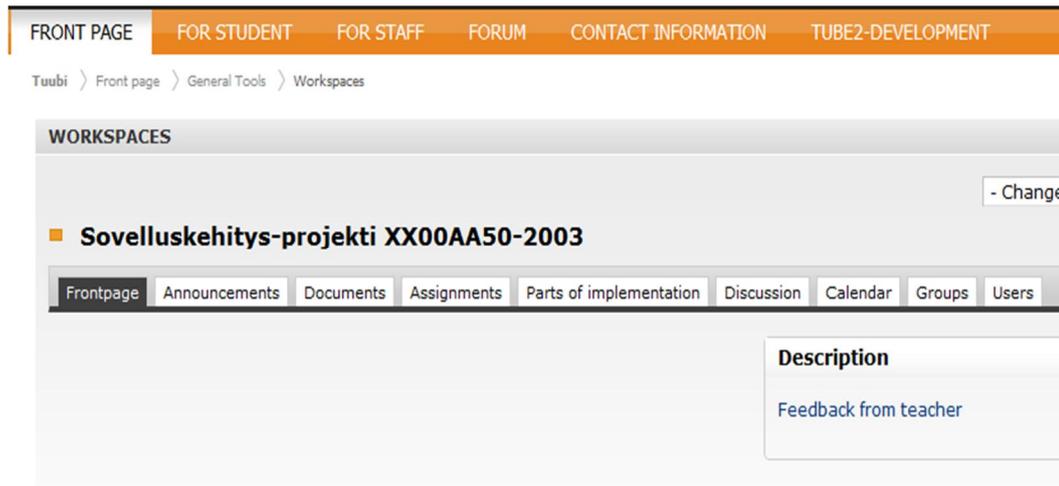


Figure 1 Metropolia University web portal

In the above figure 1 it can be seen that the student is offered an integrated learning environment. Some of the functionalities are:-

- getting notification from the instructor
- obtaining documents uploaded by the instructor
- obtaining assignments uploaded by the instructor and after performing the task given returning them to the instructor to be checked
- involving in a discussion with other students or the instructor
- checking the calendar
- checking the groups

#### 3.4 How do virtual school systems work?

A virtual school system consists of an extensive database that can be used to track all learning activities and stores them for later use. The database works hand in hand with the so called collaboration tools to keep track of events like student enrollment, documents uploaded, courses offered and so on as indicated by figure 2.

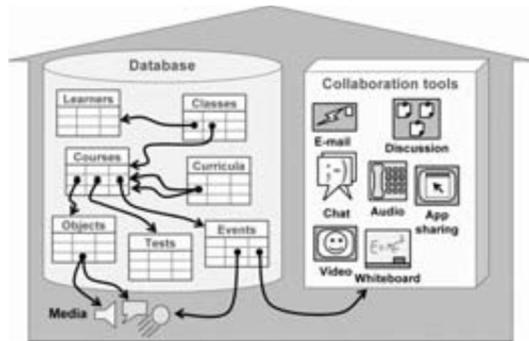


Figure 2. Database and collaboration tools in virtual school system

When we consider the courses, it might consist of tests and meeting events, media (PowerPoint presentation slides), emails, discussion forums and so on. Therefore this whole system makes a virtual school system, which instructors or students can make use of. [8]

## 4 Viope eLearning platform

### 4.1 Structure

Viope Solutions Ltd was founded in the year 2001, and offers computer programming and mathematics online, the main product being programming courses. They offer different courses which can be customized according to client's needs. The courses they offer currently are: C, C++, Java, Python, Ruby, SQL and PHP.

They distribute their product to educational organizations and private individuals, the users ranging from comprehensive school to university aged students. This system offers different parts for students where they can find the courses, the side where they can enroll to the courses offered and the instructor's side.

The Programming Courses support teachers and are also adequate for self-learning system, to access the learning system only internet connection and a web browser is needed. More than 100 educational institutions all over Finland use this service, as the following examples show:

- City of Lappeenranta, City of Imatra and City of Lahti
- The City of Helsinki Education Department
- South Carelia Vocational College
- Lappeenranta University of Technology
- EVTEK, Savonia, Laurea and HAMK

The system is available 24/7 and has multifaceted exercises. To use it there is no need to download any software, which makes it easily accessible and saves computer memory. In case of the programming tasks, automatic and immediate feedback is given to the student performing the task. The other big advantage of the system is that it is available globally for education on different levels.

The multifaceted tools provide a good control for teaching and the exercises are diverse in order to cover important parts of the theory parts, making the system efficient for contact teaching.

Course content is always comprehensive: courses include theory, editable code examples, programming exercises, multiple-choice questions, user interaction and great control features for teachers.

#### 4.2 Logging into the service & system panel

For registered users, username and password is provided in order to gain access, by using these credentials it is possible to log in to the service from all over the world. The login page is indicated by the figure 3.

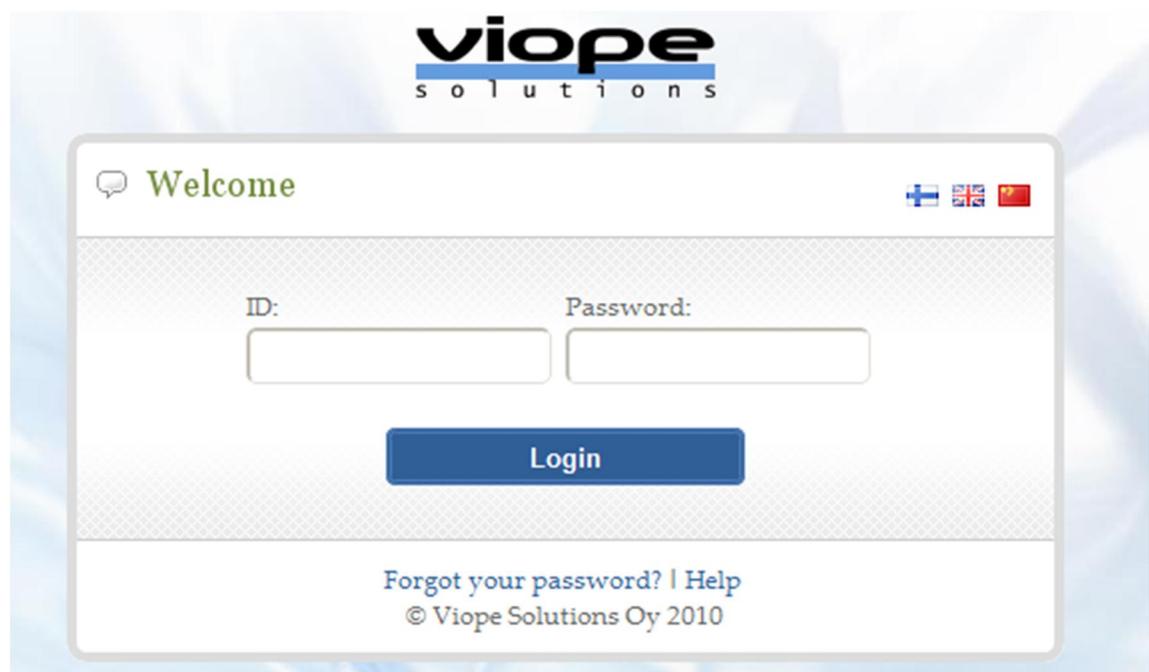


Figure 3. Login page

As you can log in to the service from all over the world, its programming environment gives you a free hand to study effectively. The URL can be customized for every customer for example school.viope.com which makes it easy to remember, connection is always SSL protected so all information is transferred safely. SSL stands for Secure Sockets Layer, it is a socket layer used for encrypting information for the secure communication over the internet.

From system panel page it is possible to select the roles of the user, according to the credential provided the user will be identified as a student or a teacher as shown in the figure 4. For a student, it will be possible to check the courses offered for him or her; and studies can be started by clicking the name of the course. It should be noted that the validity period does not necessarily correspond to the period allowed by the instructor for completing the course.

The screenshot shows a system panel with three tabs: "COURSES ON WHICH YOU ARE A STUDENT" (selected), "COURSES ON WHICH YOU ARE AN INSTRUCTOR", and "COURSES ON WHICH YOU CAN ENROL". To the right, there is a link for "Expired courses". Below the tabs, a text box explains that the list shows courses where the user's ID is accepted as a student, with validity periods shown after the course name. It notes that expired courses move to the "Expired courses" category and that validity periods may not match instructor completion times.

COURSES ON WHICH YOU ARE A STUDENT	COURSES ON WHICH YOU ARE AN INSTRUCTOR	COURSES ON WHICH YOU CAN ENROL	Expired courses
<p>The following is a list of the courses on which your ID is accepted as a student. You can start your studies by clicking the name of the course. The validity period of the course is shown after its name. After expiration, the course will be shown in the "Expired courses" category. Please note that the validity period does not necessarily correspond to the period allowed by the instructor for completing the course.</p>			
<p><b>Demokoulu</b></p> <ul style="list-style-type: none"> <li>➤ <a href="#">PHP-ohjelmoinnin perusteet (st)</a> (21.09.2005 - 30.06.2012)</li> </ul>			
<p><b>viope testi</b></p> <ul style="list-style-type: none"> <li>➤ <a href="#">Viope PHP (EN)</a> (16.09.2010 - 30.04.2012)</li> </ul>			
<p><b>Demonstration school</b></p> <ul style="list-style-type: none"> <li>➤ <a href="#">Viope Introduction to Java SE (academic)</a> (23.04.2008 - 23.04.2011)</li> </ul>			

Figure 4. System panel

And if the user is a teacher in this case, he or she will be able to access the courses on which he or she is a teacher; the validity period of the course is shown after its name. After expiration, the course will be shown in the "Expired courses" category.

On the third tab it is possible for a student to enroll for courses, the following is a list of the courses on which your ID is able to enroll. The student can enroll by selecting a course and clicking the "Enroll on the course" button. Once the course's instructor accepts student's ID for the course, an announcement of acceptance will be sent to the

email address the student provided, and the course will be shown under "Courses where you are a student".

Depending on the course there are 10-20 chapters of theory to study. Each course has enough programming exercises at the end of the chapters, which helps to assess the student's knowledge on the subject matter as indicated by figure 5. There are also multiple choice questions which are provided to enrich the student's knowledge.

The screenshot shows a web course interface. On the left is a 'STUDENT'S MENU' with a grid of chapter numbers (1-18) and a list of navigation options: Chapter 1: Theory, Multiple-choice questions, Programming exercises, Open exercises, Exams, Extra material, Communication, My corner, Help, Ask a tutor, Manual, Free practice, and Settings. The main content area is titled '1 Foreword' and '1.1 Course introduction'. It contains introductory text about the course's structure and a list of sections: 1. Foreword, 2. Algorithm, and 3. Introduction to Java. The text describes how the course is organized into chapters and includes programming exercises.

Figure 5. Theory page

In order to help the student in the theory materials there are enough code snippet examples. These examples are indented, colored and commented to give the student how it would appear in the code editor. There by helping the student to understand which part is which, and when he or she is coding would realize how the color coding for each part is and how to recognize errors.

The coding examples show also how important it is to comment a code, since it will help others and the student themselves understand what they have written. Students

often forget to comment their code, and when trying to understand it at a later time, is very difficult. Therefore it can be seen that the student is learning not only how to write a code but also the discipline how to write them, when he or she is studying the code examples provided.

All the provided programming exercises, multiple choice questions and open exercises appear on the students menu. The student can also take exams and make use of extra materials and all the functionalities available on the menu.

### 4.3 Learning material management & exam management

In the learning material management it is possible to add chapters, edit chapters, publish chapters, search for theory pages and get help topics. This editor is easy to use for teachers who are providing the different chapters for different courses. For example to add chapters there is a link which lets the user create a new chapter as indicated by figure 6, and in the chapter that is being created it is possible to create theory pages and multiple choice questions.

The screenshot displays the VIOPE PHP (EN) Learning Material Management interface. On the left, there is an 'INSTRUCTOR'S MENU' with the following items: Learning material management, Course management, Exam management, Extra material, Exercise bank, Evaluate exercises, Student followup, Communication, Help, Manual, Free practice, and Settings. The main content area is titled 'LEARNING MATERIAL MANAGEMENT' and includes a navigation bar with 'add chapter', 'Publish all chapters', 'Table of contents', 'Search for theory pages', and 'Help'. Below the navigation bar, there is a list of 'Unpublished chapters' numbered 1 to 11, each with a title and a set of icons for editing and publishing. A red arrow points to the 'add chapter' button, and a black arrow points to the icons for the first chapter.

Figure 6. Adding chapters and publishing in learning material management

The other important functionality in this management editor is that, when chapters are created they are not published, which means that they are not open for students to see. Once the teacher decides that the chapters are ready and can be put to action, the editor will allow him or her to publish it by pressing the Publish immediately icon which is shown by the black arrow in the figure 6. It is also possible to edit the publishing date by pressing the icon which says edit publishing date when mouse is hovered over it, and if it is necessary to delete the chapter created it is possible by pressing the remove chapter icon.

The screenshot displays the VIOPE PHP (EN) Learning Material Management interface. On the left is an 'INSTRUCTOR'S MENU' with options: Learning material management, Course management, Exam management, Extra material, Exercise bank, Evaluate exercises, Student followup, Communication, Help, Manual, Free practice, and Settings. The main content area is titled 'LEARNING MATERIAL MANAGEMENT' and includes a 'Back' button, a 'Theory pages' section with a list item '1. Added by teacher page 1 (Introduction Welcome to the PHP course where you w...)', and three exercise sections: 'Multiple-choice questions', 'Programming exercises', and 'Open exercises', each showing 'No exercises'. A red arrow points to the 'Back' button, and a blue arrow points to the 'Multiple-choice questions' section.

Figure 7. Learning material management

As can be seen from the above figure 7, it is possible to add theory pages and multiple choice questions for the different chapters, and after adding them it is also possible to edit them or delete them. The theory editing page is demonstrated in figure 8 below.

## LEARNING MATERIAL MANAGEMENT - Editing a theory page

[Return to editing the chapter](#)

**Introduction**

Welcome to the PHP course where you will learn the basic PHP syntax for creating dynamic web based applications. PHP is a powerful scripting language that provides dynamic functionality to the developers, operating requests and responses automatically. PHP cannot do anything alone though. It must always be embedded into HTML, therefore its called HTML-embedded scripting language. PHP is similar to C, Java and Perl but also has its own unique features. PHP is most commonly used for form processing and math calculations at the back- end level (server side), and provides the results in a browser-friendly format such as HTML pages.

**Who should attend the course?**

This PHP course requires no previous knowledge about PHP. If you have experience in similar programming languages such as ASP, JSP or C, you'll find the course much easier. The only requirement for the course is having a basic understanding of static

Path: p » strong

[Save](#) [Preview](#)

Figure 8. Learning material management panel for editing theory page

The other functionality available under the instructor's menu are the course management, exam management, extra material, exercise bank, evaluate exercises, student followup, communication, help, manual, free practice and settings.

Under the course management it is possible to manage the course bulletin board, students' enrollment and student's passwords as indicated by figure 9. In case of the course bulletin board, it is possible to add a new bulletin, modify or delete the already existing one.



Figure 9. Course management panel

The important managing feature in the course management is the enrolment management, which allows the instructor to manage students' enrollment which can be seen from figure 10. The instructor can see the name, id, student number, group code, email and date of enrolment of the student. After getting all the information, it is possible for the instructor to select a student or group of students to add them in to a course or remove them from the registration. It is also possible to see the students who are accepted to the course.



Figure 10. Enrolment management

Under exam management it is possible for the instructor to upload exams that he has provided for the specific course as indicated by the figure 11. To upload an exam to the system, the teacher chooses Add new exam then will have to provide a title for the exam. After providing the title the instructor can see the multiple choice questions, programming exercises and the open exercises he or she has uploaded to the system. And can modify, remove or add a new exam as indicated by the figure 12 and 13.

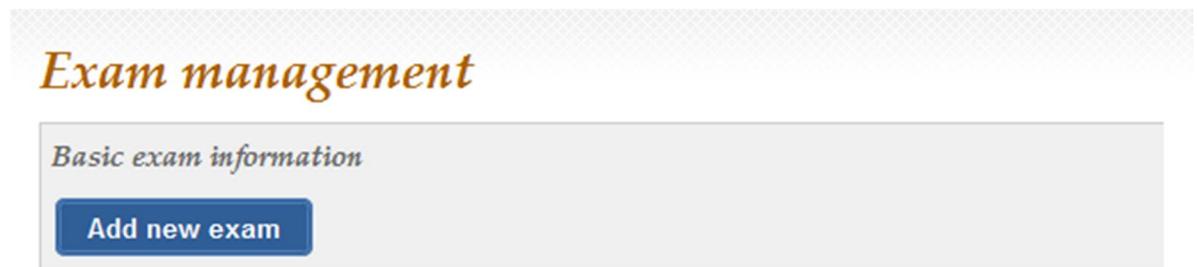


Figure 11. Exam management



Figure 12. Exam management panel for adding exam

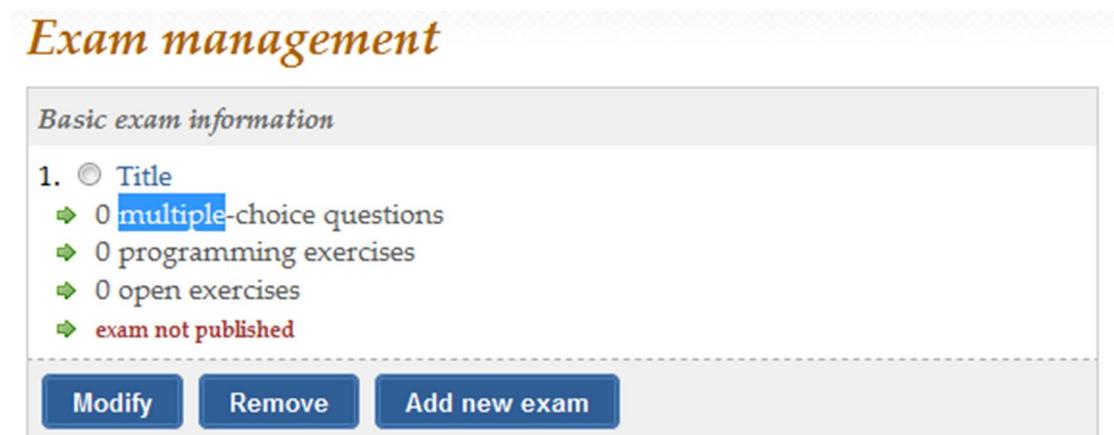


Figure 13. Exam management panel for managing all exercises

#### 4.4 Extra material management & exercise bank

In the extra material management the instructor can upload links that are related to the subject matter. This way if the student wants to find out more about the chapter, it will be possible to refer to extra materials to gain more knowledge as indicated by the figure 14. The instructor can also upload files such as pdf files or word document or even code examples, which might serve as a start up for the student for specific tasks.



Figure 14. Extra material management

Inside the list of links the instructor can arrange the links under appropriate topics, since as the number of links grows it will be very difficult to identify which link is related to the which chapter as indicated by the figure 15. Therefore it makes it easy for the student to find information.



Figure 15. Extra material links management

As a beginner in the programming world, it helps to see what other programmers have been doing and how they do it. So it is a common practice to follow a step by step tutorial given by experienced programmers. After getting the general overview, it would be a good idea to give students a startup files from which they can build their own projects.

Therefore it is an important feature to provide a place where instructors can upload files as indicated by the figure 16, which can be a startup files or other files which

might be valuable to the student. To upload a file the Files link must be clicked which in turn lets the instructor upload the files as indicated by figure 17. While uploading the file it is possible to give a description for it.



Figure 16. Extra material file management

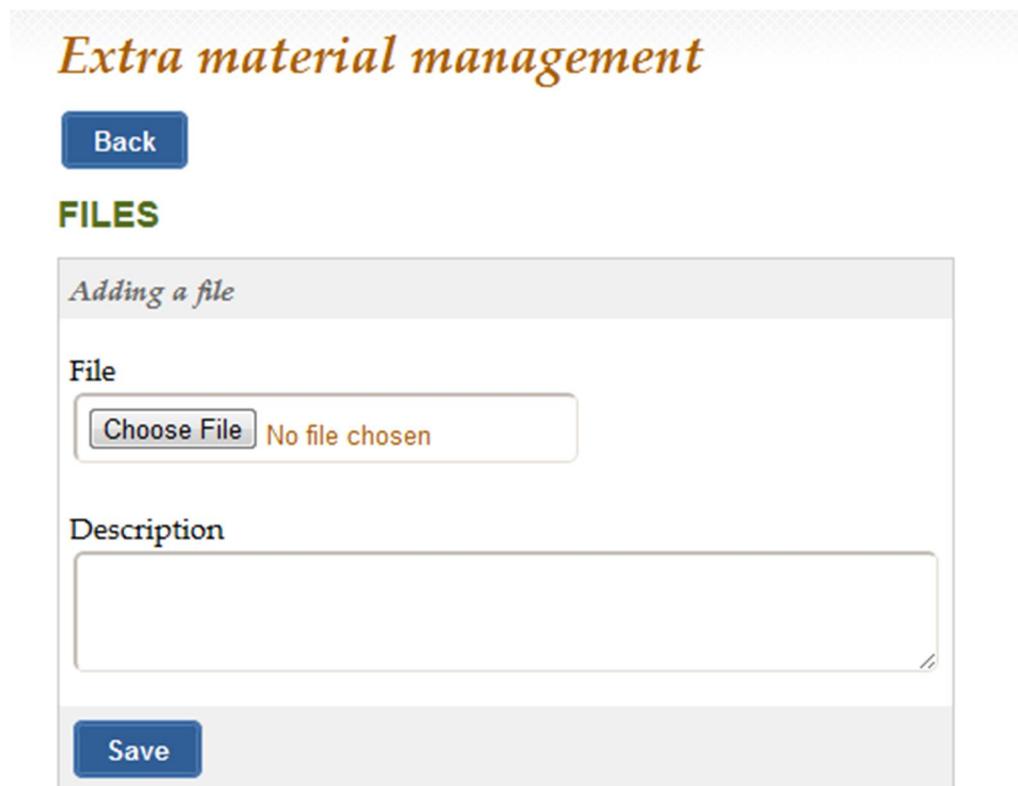


Figure. 17. Extra material management panel for adding a file

In order for the programming exercises, multiple choice questions and open exercises to appear on the exam management window they have to be uploaded by using the exercise bank as indicated by the figure 18.

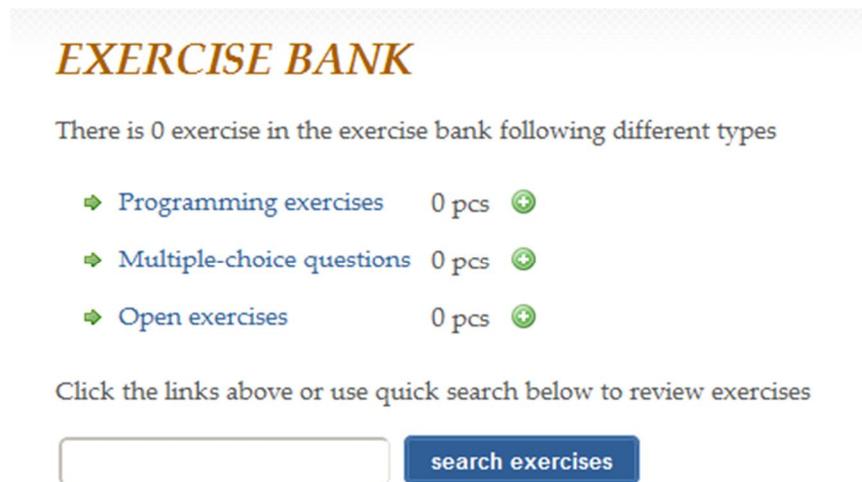


Figure 18. Exercise bank panel

These different exercises have links which will let the instructor to edit them easily. In all exercises types it is possible to:

- give a title to the exercise
- give the task to be performed by the student
- give example answer as a reference to students
- assign an author
- mark it as incomplete or published in order to indicate the status of the exercise
- search for an existing exercise
- and reset the description of an exercise if needed
- and finally adding the exercise as indicated by figure 19

**EXERCISE BANK**

[Back](#)

Search programming exercises

Title   Incomplete

Assignment   Published

Example answer

Author  ▾

[search exercises](#) [Reset](#)

The search returned 0 hits

[Add new Programming exercise](#)

Figure 19. Exercise bank panel for adding programming exercises

#### 4.5 Evaluate exercises & student followup

As an instructor it is important to evaluate the exercise the student has done and perform a student follow up, in order to keep track of the student's progress.

**Evaluate exercises**

- [Give points to students' solutions](#)
- [Modify evaluated solutions](#)

Figure 20. Evaluate exercises panel

As shown in the above figure 20, the instructor can give points to the students' solutions and modify evaluated solutions. And as shown in the figure 21 below, he or she can keep track of the students' progress by searching for completed exercises. One important functionality this system offers is the way to check plagiarism, this can be done by using the identify plagiarism in programming exercises tool. The system also has a tool which represents the students' activities in terms of charts, which helps the instructor to identify how much the student has spent time on the course.

## STUDENT FOLLOWUP

 Searching for completed exercises.

Look for students completed exercises using various search terms.

 Identify plagiarism in programming exercises.

Look for similar programming exercise solutions among other students.

 Statistics

Check out how students have spent time at the course.

Figure 21. Student followup panel

### 4.6 Communication & help

It is evident that while studying there will be obstacles which the student has to overcome by getting help from others. In order for this to work there should be some kind of means of communication. Both communication and help functionality are available in here from which the student can get help from. Under communication there are discussion forums, a place to store messages and chat functionality in order to allow whoever is in the viope system and online to communicate as indicated by figure 22 and figure 23.



Figure 22. Communication panel

## DISCUSSION

### INTERNAL DISCUSSION FORUMS FOR THE COURSE

<i>Area</i>	<i>Threads/Messages</i>	<i>Latest message</i>
<a href="#">Free discussion</a> Everything from A to Z can be discussed here.	0/0	..
<a href="#">Discussion on the Viope World learning environment</a> Ideas associated with the learning environment should be written here.	0/0	..
<a href="#">Discussion on the PHP language</a> Do you have problems with the language barrier? Help can be found here.	0/0	..
<a href="#">Discussion on the course</a> Issues and announcements associated with the course will be shown here.	0/0	..

Figure 23. Discussion forums

#### 4.7 Chapters

One of the courses provided is PHP course in English, this is the learning material that me and my group met worked on. Under this course there are 10 chapters as shown in the figure 24 below.

Unpublished chapters	
1. <a href="#">Introduction</a>	
2. <a href="#">Chapter1</a>	
3. <a href="#">Chapter2</a>	
4. <a href="#">Chapter3</a>	
5. <a href="#">Chapter4</a>	
6. <a href="#">Chapter5</a>	
7. <a href="#">Chapter6</a>	
8. <a href="#">Chapter7</a>	
9. <a href="#">Chapter8</a>	
10. <a href="#">Chapter9</a>	
11. <a href="#">Chapter10</a>	
12. <a href="#">Title</a>	

Figure. 24 Chapters for PHP course

1. Introduction
2. Chapter1- Introduction, Eclipse Installation, Syntax, variable, echo
3. Chapter2- Datatypes, Operators
4. Chapter3- Include-File, Require
5. Chapter4- If-statement, if...else, else-if, switch
6. Chapter5- Forms, Post & Get
7. Chapter6- Array, Functions
8. Chapter7- For Loop, While Loop, Do while, For each
9. Chapter8- File create, File open, File close, File Read, File Delete, File Append, File Upload
10. Chapter9- Date, Session, Cookies
11. Chapter10- OOP, NameSpaces

The look and feel of the chapters provided has the same standard to give a good outlook to the whole system and for easy understanding. The chapters are organized from basic to advanced subjects. An example of what the theory part looks like is shown in the figure 25 below.

## 2.1 PHP data types

Variables are very useful for storing different types of data. Different types of data take up different amounts of memory and can do different things. For example we use integer data type for whole number and string data type for a string of text. PHP is a **loosely typed programming language**. This means PHP will determine the data type at the time data is assigned to each variable i.e you can change a variable's data type at any time.

Table 2.1 data types example	
<code>\$type = 3;</code>	<code>(integer data type)</code>
<code>\$type .= "change me as you wish";</code>	<code>(string data type)</code>
<code>\$type = 2.35;</code>	<code>(float data type)</code>

In the above example the variable `$type` switches between three data types, First the variable holds an integer, then string and finally float data type. It is very important to keep track of the data type of the variable. For example, if the code manipulates an array variable and the variable contains a number value instead of an array structure, there will be an error when the code attempts to perform array-specific operations on the variable.

## 2.2 Scalar data types

Scalar data is data that can hold only one piece of data at a time. Below are the 6 scalar data types:

Table 2.2 Scalar data types

Type	Description	Example
string	a collection of characters	"Hello"
integer	a whole number	5, -30
float	a floating point number	15.7, -43.21
boolean	true or false	true, false
unicode	a sequence of unicode characters	"Hello", "abc123@# \$"
binary	a sequence of binary (non-Unicode) characters	"Hello", "abc123@# \$"

Fig. 25 Theory page

As can be seen from the figure 25, every chapter has a well explained theory part supported by examples. In the above chapter for example it has 7 pages, which students can read and give their feedback on each page. This is possible by clicking the "Give your feedback about this theory page" link. After reading and checking the examples given, the student then is expected to do the multiple choice questions provided.

## 5 Eclipse plugin for Viope system

### 5.1 Accessing the system

The factor that directed Viope system owners in to the need to develop an eclipse plug-in is the fact that many institutes are using eclipse editor for programming their applications.

This editor is a platform which can be extended and used for the construction and integration of software development tools. The dependency of the developers comes from the fact that, Eclipse as a platform provides components which are open source softwares. These softwares help in the development of solutions so called plugins, which can be integrated with workbenches.

In Eclipse reusability of tools is regarded as one of the qualities of the editor. But it is a fact that not all of the codes provided publically are to be trusted. Therefore it takes a bit of effort from the user to verify the reliability and use them as a starting point.

Since there are a lot of tools and API's provided over the net, it is very hard for developers to choose from all these tools. But Eclipse has a reputation that precedes it, which leads to being trusted by developers worldwide; thereby strengthening the trust and number of Eclipse user community. Since the source code for this platform is provided developers can have confidence on it. In addition it gives them an advantage to work with the APIs which are open for them.

As the saying goes "two sticks are better than one", the collaboration of developers while working on the same project yields a much better output. Therefore it is easy to see how this platform helps developers to work together.

Clearly stated code with the supporting documents, help developers understand the code easily provided by others. So even though somebody out there provides the source code, it will be easier for the developer to understand the code and implement it since the code is well documented

The presence of collaborative discussion forum and a huge society working on this platform makes it possible to get answers to problems that developers face especially

when they are beginners. Since there is almost always some other developer which has faced the same kind of bug as the others, they can share their experiences and solve the problem easily and faster.

In this platform there is a long term support provided for whoever needs it, and makes developers comfortable and depend on it. Because of this reason the editor acts as a safe net for small companies to grow and sustain their business for a longer period of time.

Flexibility is the other flavor of Eclipse platform. The platform gives the chance to modify the components according to the need of the developer. For example, if the editor is inadequate, a plug in can be developed to extend it. This is actually what we did in our project.

In a nutshell Open source, an open community and an open platform create an environment, where the needs of tool builders in building small or large projects are satisfied. [9]

The advantages stated above clearly shows how popular and reliable the Eclipse platform is, and makes it clear why it was chosen to deliver a good way of teaching students. As mentioned above to integrate the Eclipse platform and the system, a plug-in was developed. Next step would be explaining what the general definition of plug-in in Eclipse is. The purpose here is not to describe how a plug-in is developed in Eclipse, instead to give a brief introduction what a plug-in is and what kind of plug-in was developed. [10]

Eclipse plugin is a jar (Java ARchive) file, with certain services and which can be integrated with the current Eclipse workbench. This means that the plug-in developed and implemented does not affect the other already working ones, this avoids the risk of crashing the whole system and enables different kinds of plug-ins to be integrated together.

After plug-ins are developed, they would have to be deployed before being put to use. This is done by choosing the export wizard on Eclipse editor, and then choosing the plug-in to be deployed and giving a location to where it should be deployed to. This file will be stored under Eclipse directory in a folder called plugins. Then this jar file must be placed in a folder called dropin under Eclipse folder.

Now it would be time to look at how we used Eclipse in to our advantage. To access viope's system the student or teacher should be a registered user, hence will have a username and password. In our plug in there is a login window, which the users can use to login. This is very important because this is the only way to identify the user, since the user at this point is not logged on to the system. When the user provides the credentials the session id will be stored which will be the key to getting the chapters and exercises desired.

For developing the login project the Standard Widget Toolkit commonly known as SWT was used. Standard Widget Toolkit as the name implies is a toolkit which is used to display graphical user interface elements, in addition this toolkit is implemented only in Java platforms.

As can be seen from the code in Appendix 1 the shell object is used, the shell object is the highest-level container, which contains other widgets (e.g. buttons, labels etc.) inside of it. By using this object it is possible to set the layout, set the text and so forth. In our case the layout and text are set as shown above. The above code gives a window where students can use to login to do their exercises.

When the user logs in by using the username and password, he or she will be identified by the plug-in which in turn allows him or her to access the chapters and exercises provided. More explanation is given in the following topics.

### URL Encoding

Uniform Resource Locator commonly referred to as URL is a unique address given to a document on the internet. Uniform Resource identifier also known as URI is an identifier used for uniquely denoting name or resource on the internet. The encoding of information being transferred by using URL is called URL encoding.

The format known as ASCII is the way how to send URLs over the internet. But since characters which do not correspond to this format are mixed often in URLs, the URLs must be encoded.

In order to say the URL being sent is encoded, the characters being sent should be converted into a standard format. Since characters like "#", "/", "." might be included in the URL which are susceptible to alteration, they should be encoded.

For example, the "#" character is an html anchor which means that it is a reserved character. Another example is the space character; since it is not allowed to have a space character in a valid URL format it must be encoded. Also other characters like "~" which when is used in URLs on the world wide web denotes a personal website on a Unix based server, might not be transported properly across the internet so we should encode it.

In order to avoid all this alteration of the data transmitted over the internet, URL encoding replaces unsafe ASCII characters with a "%" followed by two hexadecimal digits. In case of spaces in a URL, the URL encoding replaces a space with a + sign. As a whole it ensures the conversion of string into valid URL format, which are called "alpha | digit | safe | extra | escape" characters.

This brings us to the question what should be URL encoded? The answer would be any alphanumeric character. In our case, the student needs to provide credentials in order to login and start working on the programming tasks. And the username and password the student provides must be encoded in order to protect these credentials from falling into the wrong hands. The code snippet in Appendix 2 shows how that was achieved.

[11,12]

### Reading from and Writing to a URLConnection

We have seen what a URL is and what is URL encoding and why we encode. The next step would be to make a connection with the server in order to get all the needed ids from the xml response. This xml response is sent back as a response to the students request to login, and if the student is a registered user the server will give an xml response which has the session id, the chapter id and the exercise id. These ids will be discussed later in the coming topics.

The main focus here is how connection is achieved by using URL connection class. By using this class it is possible to get methods which will allow a user to communicate with a server, but these methods prove to be useful only with HTTP URLs.

### Reading from a URLConnection

After connection with the server has been made by using URLConnection, the next process would be to read from it. The objective here would be to get an object instead of directly accessing the input stream. It would have been possible to connect to a server by using URLReader, but for our case URLConnection was chosen because by using URLConnection it is possible to write to a URL as well. [13] The code is shown in Appendix 3.

The reason why it is needed to get the stream from the URLConnection is that, communication with the server does not start even though we have retrieved the URLConnection object using `openConnection()`. It is also possible to add or change headers and other connection properties before opening the connection after establishing it.

Lifecycle of the URLConnection would now be first to get an input stream, after that it is possible to use the GET method, sends the headers to the server and also read the output from the server. And when an output stream is fetched, it is possible to send it using POST method. It is necessary here to note that calling the method `setDoOutput(true)` might be needed, to let the program understand that we are writing data to it. And when the input stream is fetched, the output stream is closed and while waiting for the response from the server.

The disadvantage of using a URLconnection is that, once a server specifies a content length then URLConnection will keep the underlying input stream open until it receives that much data, even if explicitly closed. Therefore if it is needed to shut a client down for some reason, it would not be possible to do that since URLConnection would keep the network connection open.

The usage of the method `close()` on the `InputStream` or `OutputStream` of an `URLConnection` is as the name implies to close the connection made between an instance and the server. It should be noted here that, even though it is possible to make a single request by using `URLConnection` instance it is also possible to share the underlying network connection to the HTTP server with other instances. Another way of disconnecting the connection is by using the method `disconnect()`.

After opening the connection, `StringBuilder` is used for inserting and appending data. What is meant by appending is to add characters on the already existing ones, while inserting performs the task of putting new data to a specified place.

For example, if we have a set of string characters like "Metro", and if we wanted to implement the append method of `StringBuilder`. First there must be a `StringBuilder` object which contains the string, let us call it X. If the string "polia" is needed to be appended, we do it by calling the append method like `X.append("polia")`. The append method will add it to the first string and the output will be "Metropolia". [14]

In order to read fast from an input line, the use of `BufferedReader` is imperative. By using this object each line of input are converted into a java string. When the `BufferedReader` `readLine` method has finished reading it will return null. The example below shows how while loop iterates through the buffer read `readLine`. [15]

```
while ((line = bufferedReader.readLine()) != null)
```

## 5.2 Data translation and transformation

Integration of environments is the important aspect in today's enterprises, but in reality applications in the enterprises have their own data format. In order to tackle this issue there must be a way to enable integration of different application; otherwise it would be a difficult task to make the business system efficient.

This makes it evident that the data received by the system must be translated or transformed into the appropriate format before being implemented. Here is where XML comes in to action, which facilitates the integration of data across different systems and it is being adopted by different business sectors for a standard format for information exchange.

Using this standard has made it possible to simplify both intra and inter enterprise integration. In order to exchange data XML standard format can be used, even though the original data was created in a different format. All that is needed is to change the current format in to XML standard format. Just like any other language XML has a grammar, and it is possible to further transform XML document in to this grammar and exchange efficiently over the internet between trading partners.

There are many tools which provide support for the translation of legacy data into XML and transformation of XML documents into other XML grammars, one of which is Web Logic. This logic creates a single flexible environment by intergrating components of business. For our discussion here we would not go into this, instead see the general description of what XML is and how it is parsed. Since for our project what we need is some information from an XML response, we will see later how to parse an xml document.

So before we proceed any further let us see what xml really is and what we can do with it .XML stands for eXtensible Markup Language, and by using this language it is possible to transport and store data and it provides a platform neutral and language independent way of describing data. Xml is similar to HTML (hypertext markup language), but unlike HTML XML does not consist of fixed markup tags. Instead it allows you to define your own set of tags hence the name extensible.

Some of the advantages of XML are listed below:

- Universal data format for integrated electronic business solutions
- Self-describing data
- Complete integration of all traditional databases and formats
- Modifications to data presentation needs no reprogramming

- One server view of distributed data
- internationalization
- Open and extensible
- Future oriented technology

Next step would be to see how XML document is structured; because when we see how we are reading xml later we will need to know how the document is structured and access points of the different elements. An XML document is a hierarchical structure called an XML tree, which consists of nodes of various kinds arranged as a tree.

The top most node is called the root. The nodes in a tree may have different numbers of children, and nodes with no children are called leaves. Let us see one example of an XML document.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
  <to>viope</to>
  <from>Student</from>
  <heading>Registration</heading>
  <body>I want to register to your system</body>
</note>
```

In the above example the first line is the XML declaration, which defines the XML version in this case 1.0. And it tells also the encoding used (ISO-8859-1 = Latin-1/West European character set). Note here that the other tags are user defined not a fixed tag by XML rules. [16]

### 5.3 SAX parser

There are a number of XML parsers of which we can mention SAX and DOM. When we look closer to these two parsers, we can see that the need for a common API for all event based parsers led to their introduction. SAX is an API to be implemented by event based XML parsers, how it works is that as this parser moves through the docu-

ment, events such as the beginning of an element or the end of the document are reported via callbacks to an event handler. And since it has an object that can fire events, the listeners are able to handle those events. These listeners implement an interface that the firing object understands, and then register with that object as being able to handle the events generated.

So the method defined in the interface will be called once an event occurs and object has notified each listener registered to handle that event. In SAX, when it encounters the beginning of an element the event is passed along to the registered object to handle.

Below some examples SAX parsing are shown:

```
<?xml version="1.0"?>
<xmlDocument>
<title>Test SAX parser </title>
This is an example.
</xmlDocument>
```

And how SAX parses this XML is by firing the following events:

```
startDocument()
startElement(): xmlDocument
startElement(): title
characters(): Test SAX parser
endElement(): title
characters(): This is an example
endElement(): xmlDocument
endDocument()
```

Note: SAX parsing is serial, which means that there is no way to determine parent/child relationships. [17]

## 5.4 DOM parser

The other XML parser is the Document Object Model commonly referred to as DOM parser as mentioned above. One may ask since we have SAX XML parser why do we need another one?

The reason for this is that, even though SAX provides a simple way to write a program that can be used to parse an XML document when the data is read, this parser is good only for managing a few part of a document when the whereabouts of this part is known. Therefore by using SAX it is not possible to programmatically access the entire document in a non-linear order.

In order to tackle this shortcoming DOM parser is used, which is a specification for a set of interfaces that XML parser vendors can implement in order to provide a model of an XML document as a set of objects. A good definition for DOM might be, an Application Programming Interface (API) for HTML and XML documents. It is important to note here that DOM is not an implementation rather a package of Java interfaces defining the API.

Using the DOM it is possible to build, navigate the structure, delete, modify or add elements and their content. It is also possible to create, update, read or delete XML documents, by using the object Model: processing instructions, comments, entities, DTDs, and whitespace. DOM has two levels DOM level 1 and DOM level 2, DOM level 2 being an extension of DOM level 1. But since for this project the DOM level 1 is used, we will focus on this level. [18]

## 5.5 Accessing chapters and exercises

As we have seen from the previous topics we now have the tool to parse the XML response we get from the server which will allow us to access the chapters and exercises. After making a connection and reading the content that is coming from the server, it is important to know exactly the data that was needed. In our case we needed the different ids like session id, course id, chapter id, exercise id and the serial number of the exercise.

In order to get these values, it was imperative to filter out the data that was retrieved from the server. The data that was retrieved had unwanted lines, and in order to get the ids it was only needed to get the xml response from the whole response. In order to do that substring was used.

Substring is a way of clipping the important part from a response. Unlike other languages, in java it is possible to specify the end points of the part that is needed instead of specifying the length of the substring. The disadvantage of specifying the length of the substring is that it might be impractical to count the number of strings when the document it is big. But in case of java, the starting and the ending points of a data needed can be specified.

It might be a bit confusing to understand how the substring works, because it has a peculiar way of labeling the characters in the string. The labeling process starts by labeling the first character of the string 0 and the end point goes one step beyond the last character. This means when the end point is given, we give the last part which should not be included in the data. So the system will include or provide the data starting from the beginning point and excluding the last end point characters.

As an example it can be seen as , if we need to a get a data which is the English alphabet and we specify the starting point to be the character A and the end point being Z. In the output we get the characters from A to Y, and Z will be excluded from the output.

The way how substring works is that, it creates a pointer into the original immutable string. When it is said immutable string, it refers to a string whose state cannot be modified after it is created. This pointer points to the char[] value of the base string, and counts how long the base string is and extract the needed data starting from the starting offset. [19]

The methods that allows us to get certain specified characters are indexOf() and lastIndexOf(). How we can extract the needed data by using indexOf() method is by searching for the first appearance of a character or substring. And in the case of

lastIndexOf(), it searches for the last occurrence of a character or substring. The syntax for the two methods is:

```
int indexOf(int char)
int lastIndexOf(int char)
```

By using the above syntaxes it is possible to search for the first and last occurrence of a character respectively. In the above syntaxes char represents the character being searched. The syntax below shows how to get the different occurrences of specified substrings, the occurrences being first or last respectively.

```
int indexOf(String subStr)
int lastIndexOf(String subStr)
```

In the above syntaxes subStr is representing the substring. It is also possible to specify a starting point for the search using these syntaxes:

```
int indexOf(int char, int startIndex)
int lastIndexOf(int char, int startIndex)
int indexOf(String subStr, int startIndex)
int lastIndexOf(String subStr, int startIndex)
```

Next step would be to see how it was done to serve our purpose. As can be seen from below the information needed is being clipped, the information needed being a well formed XML file. So the substring starts from "<?xml" and reads until it reaches "</textarea>" which will not be included in the output because it is not part of the xml file. The next line of code checks if there is an entity "&" present in the xml file, and replaces it with an empty string. It should be noted here that if the entity is left without being replaced there will be an error generated, because it will not be read by the parser. [20,21]

```
String sb1= sb.substring(sb.indexOf("<?xml"),sb.indexOf("</textarea>"));
```

```
String nohtml = sb1.toString().replaceAll("&", "");
```

The next line of code creates an instance of the DocumentBuilderFactory. This factory allows users to get a parser object from XML documents. After creating the DocumentBuilderFactory it is turned into an in-memory object, which then is normalized. We have seen in the explanation of XML document how it was structured, using that knowledge it is easy to understand how the file is being parsed. Since there are nodes in XML files we choose the one that we want, in our case "exercises". This gives us access to the elements under that node, the needed element being "exercise\_template\_id". If other elements under the node were needed, it is possible to iterate through them by using for loop and get them one by one. [22] Code is shown in Appendix 4.

After getting the important information which is session id, the next step would be to save it to an XML file. Because if the user is not a registered user or if the user is a registered user and has not logged in, he or she should not be able to access the chapters and exercises. When the registered user loges in, the session id is written to an XML file which will be used later to access the chapters and exercises. The code below helps us achieve in writing this information to the file. Code is shown in Appendix 5.

What the above code does is that, it first creates XMLOutputFactory. XMLOutputFactory is used for getting XMLEventWriters and XMLStreamWriters. Then it creates XMLEventWriter and EventFactory, which are used to write events and creates an empty event template database respectively. Then it creates the tags needed to make up the XML file. Then it will be time to create the nodes with their respective contents, while handling the exceptions. Then the name of the xml file is assigned, so that it can be saved in a directory and provided that everything went well the final line of code outputs the "your file is written" message. [23]

## 5.6 Sending programming exercises

After having completed the programming exercises, the student should be able to send it to the viope server for checking. This will also help to keep track of the Student's progress in the particular course he or she is involved with. To send the code the student has written the following code was written, the code is shown in Appendix 6.

In the first line the user's current working directory is fetched and assigned to a string. In order for the student to send the programming exercise that he or she has been working on, the file must be chosen from a list of files.

The next line of code allows the student to do just that. `JFileChooser` presents the student with a dialog box, from which the student can choose his or her files from a directory. And the next line of code gets the absolute path of the file that the student has chosen a stores to a variable. Then this file is passed to the `FileUtils`, so that the file's entire content can be read and stored to a string. The purpose of reading the text and saving it into a string is for sending the text through the URL.

Provided that the user has done the exercise and chosen it from the dialog box, the next step would be to send it. Before sending the text it must be encoded, since it might contain irregular characters.

Otherwise the reader which is receiving the code written by the student, might interpret it in the wrong way which gives always the "you are wrong" response from the server. The last part would be reading the response sent by the server, this process is explained well in the previous section which dealt with the topic of how to communicate with the server.

## 6 How the plug-in works

### 6.1 The first proposal for the application

The programming tasks provided by viope's system are quite efficient for learning. How it works is that students are provided the tasks after the end of each chapter, the number of programming exercise depends on the chapter but on average every chapter has about 3 tasks. Currently this system has its own editor, where student can read the task and write their solution on this editor. After this is completed the system will check the code and will give the appropriate result, whether the student is correct or not. As was discussed in the previous section, the need to use Eclipse as an editor was imperative as compared to using the inbuilt editor.

In order to achieve this, the first proposed ideas where as shown on the figures below.

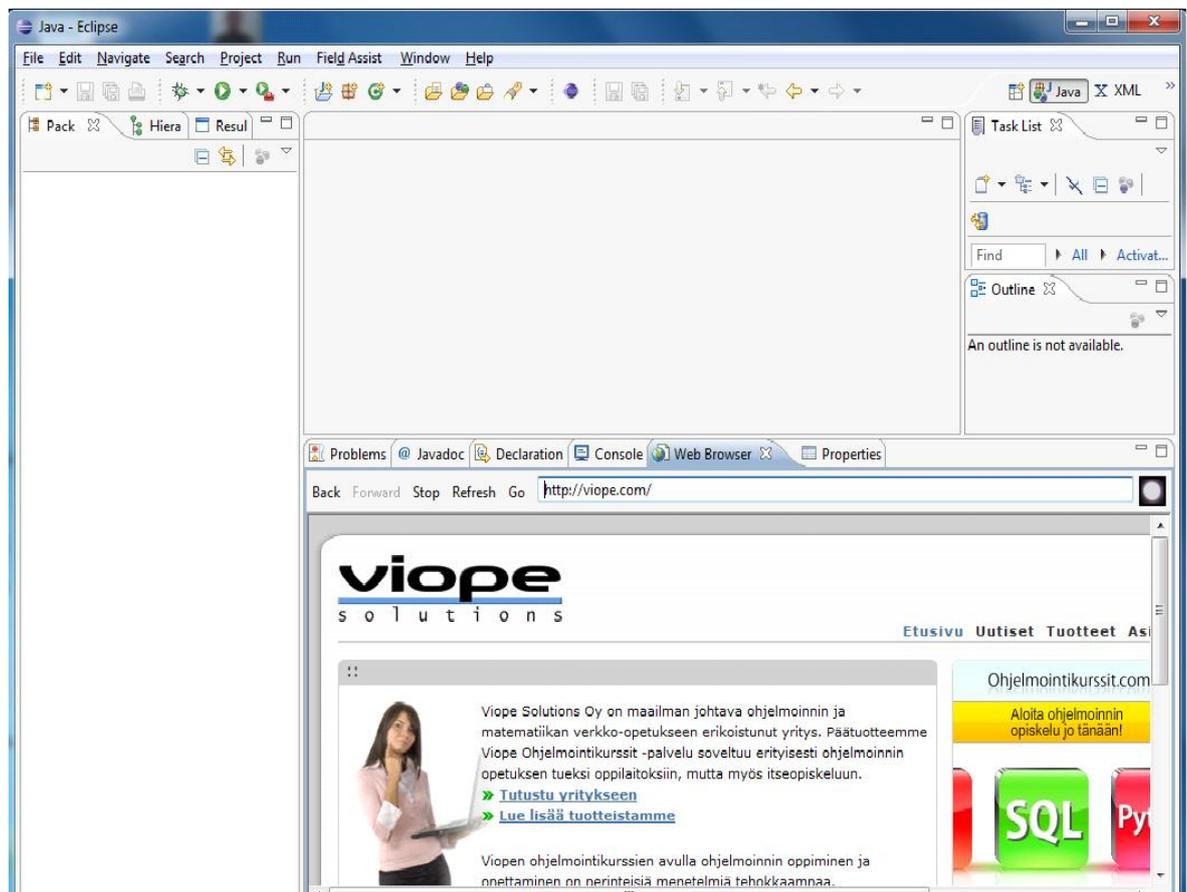


Figure. 26 Eclipse with the internal browser opened

As can be seen from the figure 26, Eclipse has different parts like the project explorer, task list and the area where it is possible to have a web browser. So the first proposed idea was to have a web browser open automatically for the students taking the courses. And the default URL would be viope's home page address.

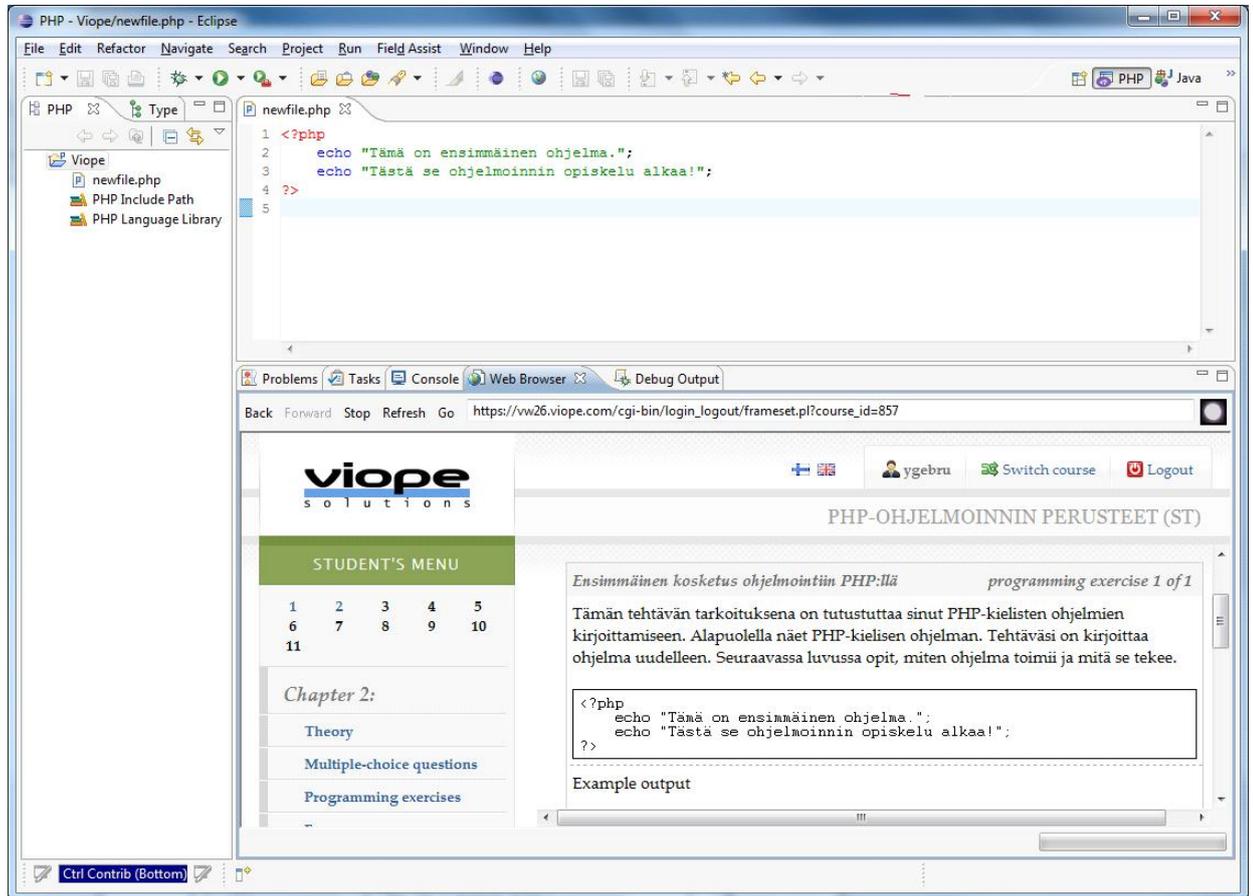


Figure. 27 viope site opened inside Eclipse

The idea of having the internal browser open was to avoid the need to open an external browser, which saves the student from switching between windows. As demonstrated in the figure 27 above.

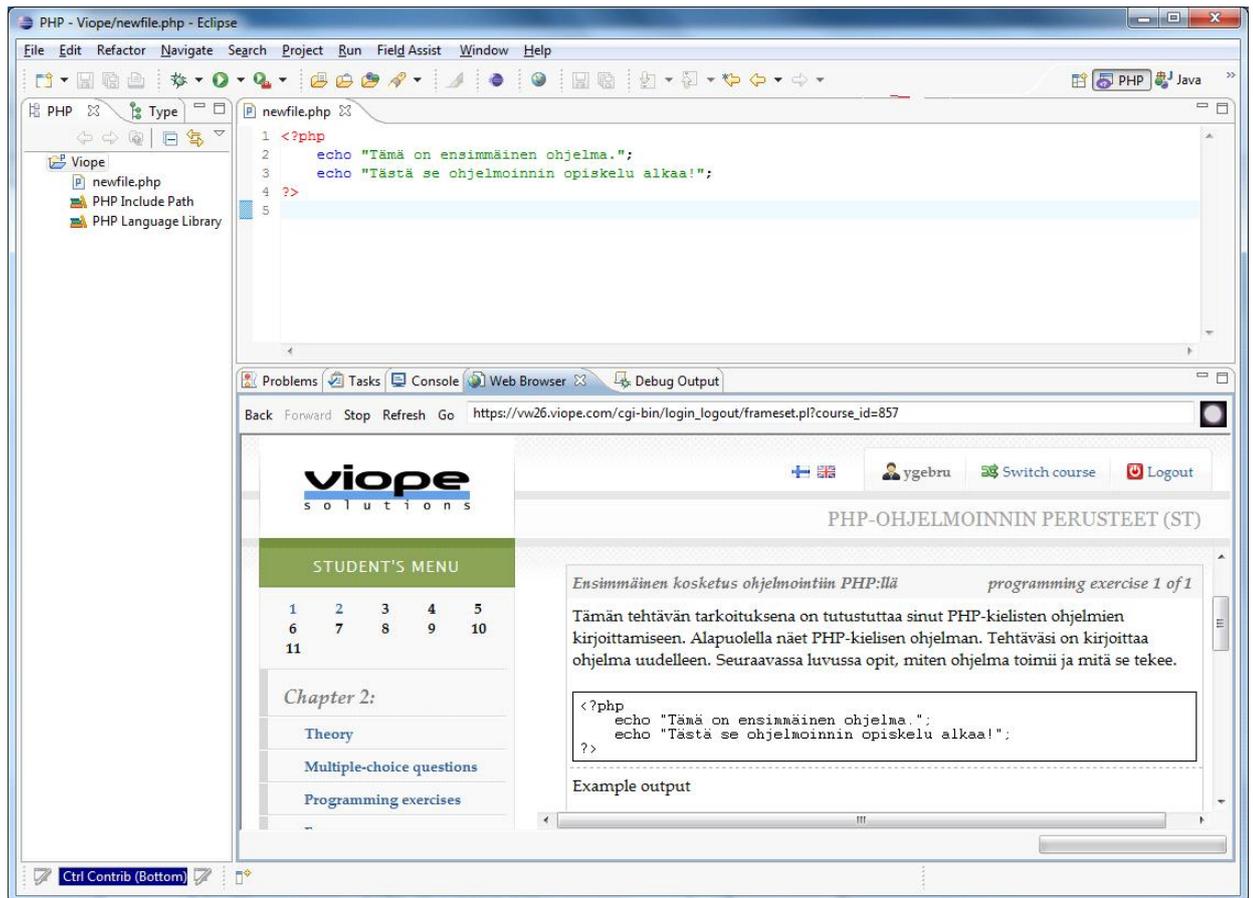


Figure. 28 using editor and internal browser of eclipse

So as can be seen from the example given in the figure 28 above, the student will login using the internal browser and using his or hers credentials. After navigating to the programming task of his or her choice the student will be able to see the task description while still being in Eclipse editor instead of navigating to external browser.

After reading the instruction given by the task, the student can create a project in Eclipse and gives the solution on the editor part. Hence the student will be aided by the editor while writing the code. All the syntax errors will be indicated and different kinds of colors appear to identify the different parts of the code. For example the opening and closing tags for PHP are indicated by red colors, while strings are indicated by green color. This kind of color coding system helps the student identify which part is which.

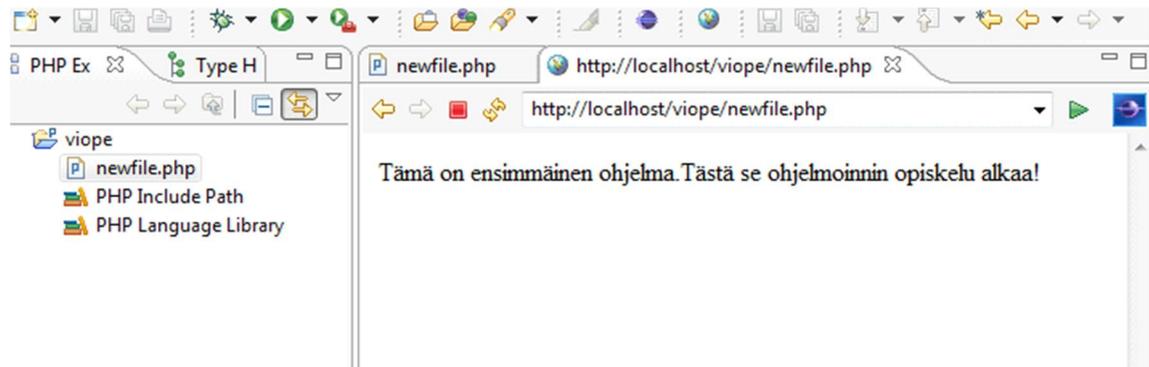


Figure. 29 Response from Eclipse editor

Then after completing the exercise when the student wants to check the answers, he or she can check them first by using Eclipse editor as shown from the figure 29 above. While trying to check the answers locally, the student must have a local server installed like apache or Wamp. The reason for the student to check the answers locally was just that not to send every answer they come up with to the server. Instead after completing the task, the student checks his or her answer by using the Eclipse editor and after getting the response from the editor sending it to viope's server for conformation.

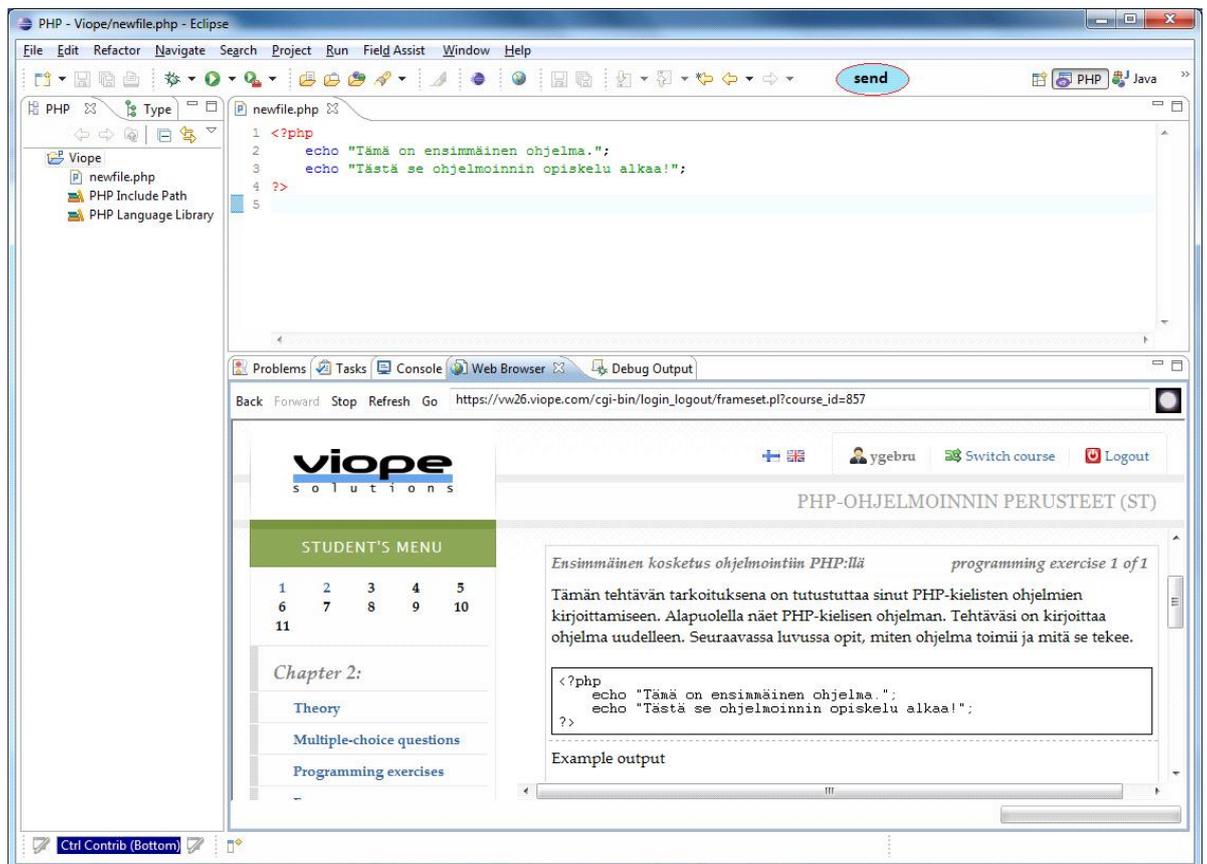


Figure. 30 Send button in Eclipse editor

It might appear that it is a repetition to check in the Eclipse editor and then check again in the server. But the reason for this was that, eclipse does not recognize the student's id since no identification is provided. The student must send the solution of the task performed to the server for registration purposes. In the viope system since the student is recognized the solution will be stored or the points will be stored under his or her name. The other reason was not to burden the server, by sending every possible solution the student can come up with thereby crowding the traffic to this server. In order for the student to send the solution to the server a button called send was provided as shown in the figure 30 above.

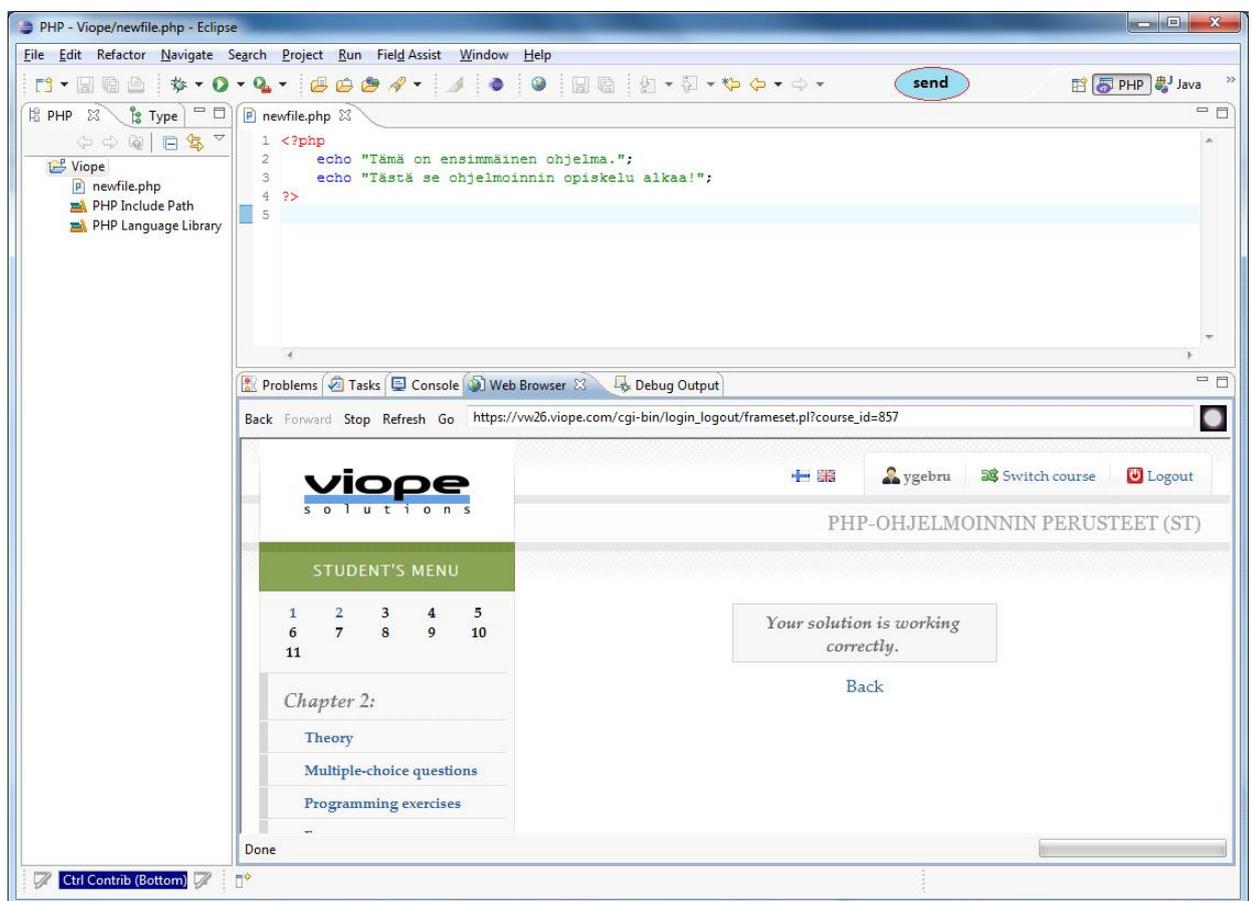


Figure. 31 Response from viope server

As shown above in the figure 31, after the student has submitted the task to the server the internal browser of the editor will show the response from the server.

## 6.2 The application done

As discussed before to use the application, the student must be a registered user and will be provided a login window. To get the login window the button is provided on the tool bar as shown with the arrow in the figure 32 below.

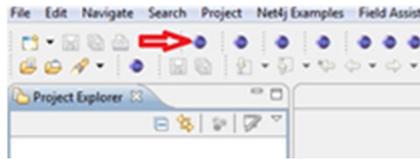


Figure. 32 Login button

When the login button is pressed the student will be provided with a login window which will ask the username and password as shown in the figure 33 below.

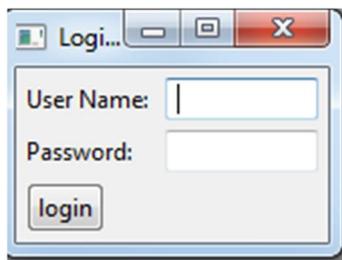


Figure. 33 login window

If the user gives the correct username and password, connection to the server will be made and the unique session id is fetched. Why this is done is explained the previous sections. And the student is shown welcome text as demonstrated in the figure 34 below.

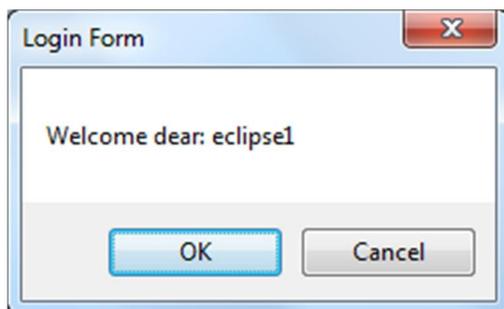


Figure. 34 Welcome text

After logging in the next thing for the student to do is to choose an exercise, this is possible by choosing the courses menu – course1 – Chapter – Exercise, from the menu bar as shown by the arrow in the figure 35 below. The need for getting the exercise in

here is for the sake of getting the needed ids so that it would be possible to send it later, not for getting the programming exercise content.

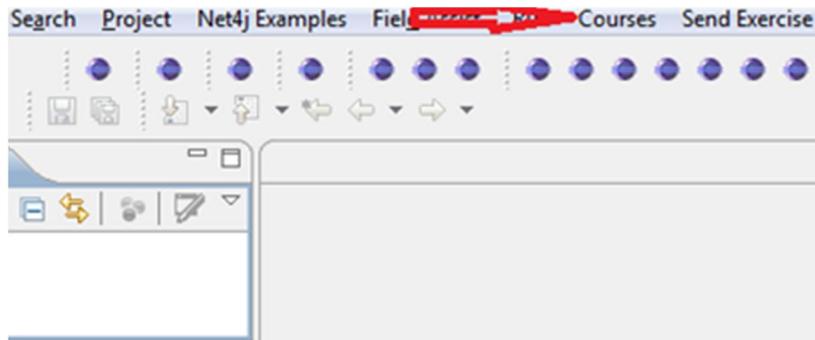


Figure. 35 Courses menu on menu tool bar

Now the student is ready to start working on the exercise of his choice. The student can get the content of the programming exercise from viopé's site. And can start working on it in Eclipse editor. After finishing the exercise the student can check the code in the editor before sending it to the server. After being satisfied with the response, the student can send the exercise by pressing the send exercise button on the menu bar shown in the figure 36 below.

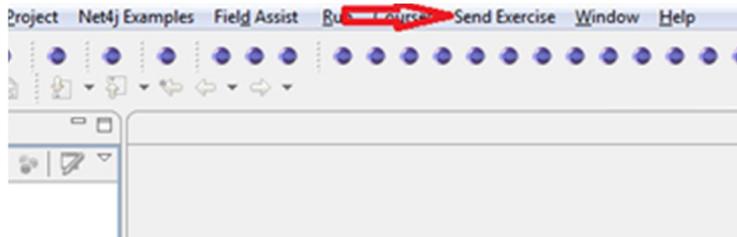


Figure. 36 Send exercise button on menu bar

When the send exercise button is pressed a dialog box which allows the student to choose the file that he or she wants to send is opened. After the student selects the file he or she is able to dispatch it to the server which in turn gives a response whether it is correct or not. The figure 37 below shows an example of a dialog box being displayed after the send exercise button is pressed.

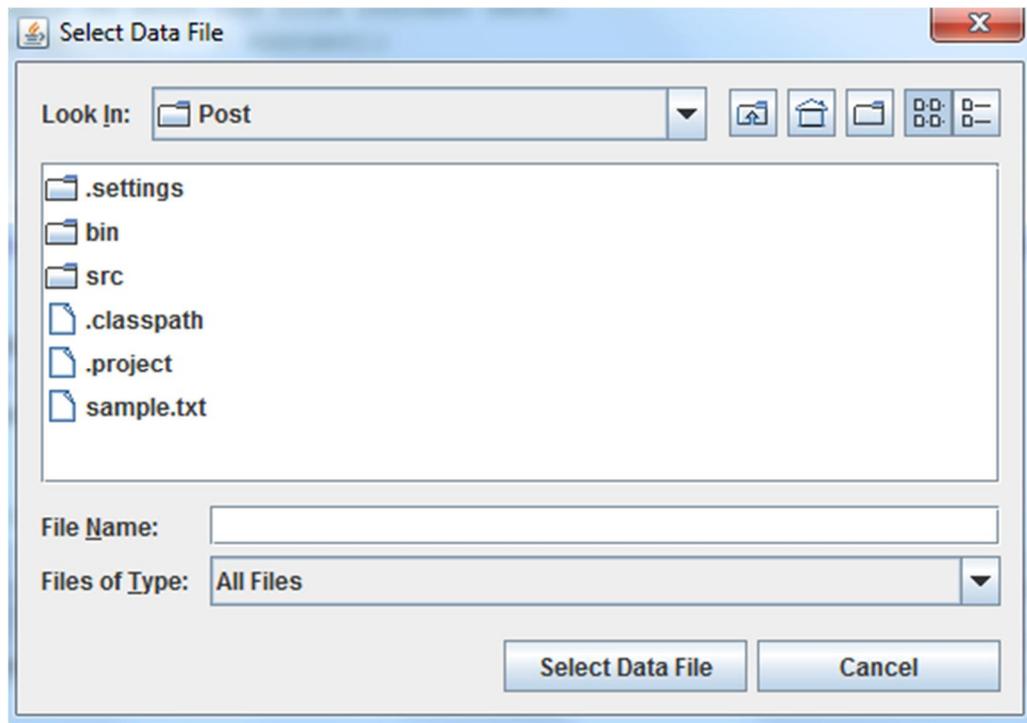


Figure. 37 Dialog box

## 7 Conclusion

The above discussion has gone through in brief the overview of E-learning system as a whole, the current viope E-learning system, the PHP learning material that was provided for this system and the Eclipse plug-in that was developed for this system. All in all the aim of the project was to get an additional course material to the system and establish a connection between viope server and Eclipse editor, which were successful accomplished.

Providing good course materials for students to develop their skills is the main goal of Viope E-learning system. To facilitate the learning system plug-ins were developed, which give extra functionality to the system. This project demonstrated a way how to connect an editor with a server and why it was needed to do so. The project done not only improves the company's reputation but is also a great tool for students who will use Viope E-learning system.

Therefore it can be deducted that we have successfully achieved our goal, which was to use the Integrated Development Environment Eclipse (IDE) in order to increase the quality of learning sessions in Viope's platform.

## References

1. Michael. Ittoolbox popular question and answer team.  
<http://knowledgemanagement.ittoolbox.com/documents/the-evolution-of-elearning-15470>. Accessed 10 January 2011.
2. Berman p. E-Learning concepts and techniques.  
[http://iit.bloomu.edu/Spring2006\\_eBook\\_files/chapter1.htm](http://iit.bloomu.edu/Spring2006_eBook_files/chapter1.htm). Accessed 12 January 2011
3. Ageless learner. October 15, 2006.  
<http://agelesslearner.com/intros/elearning.html>. Accessed 10 January 2011.
4. E-learning essentials. WorldWideLearn.  
<http://www.worldwidelearn.com/elearning-essentials/elearning-types.htm>. Accessed 15 January 2011.
5. Practical guide to e-learning for industry. Australian flexible learning framework.  
[http://industry.flexiblelearning.net.au/guide/1-5\\_Types\\_of\\_e-learning.htm](http://industry.flexiblelearning.net.au/guide/1-5_Types_of_e-learning.htm). Accessed 16 January 2011.
6. About e-learning.2007. <http://www.about-elearning.com/e-learning-advantages-and-disadvantages.html>. Accessed 16 January 2011.
7. Horton W, Katherine Horton K. E-learning tools and technologies. Wiley Publishing, Inc, pp.3-10.
8. Horton W, Katherine Horton K. E-learning tools and technologies. Wiley Publishing, Inc, pp.253-263.

9. What is Eclipse, and how do I use it?. IBM.  
<http://www.ibm.com/developerworks/opensource/library/os-eclipse.html>. Accessed 10 February 2011.
10. Bolour A. Bolour computing. July 3, 2003. [http://www.eclipse.org/articles/Article-Plug-in-architecture/plugin\\_architecture.html](http://www.eclipse.org/articles/Article-Plug-in-architecture/plugin_architecture.html). Accessed 12 February 2011.
11. Kalin M. Java Web Services: Up and Running. O'Reilly Media, Inc., February 2009, pp.176-178.
12. Permadi F. Introduction to url encoding/url encoded strings.  
<http://www.permadi.com/tutorial/urlEncoding/>. Accessed 25 February 2011.
13. The java tutorials. Oracle. 1995.  
<http://download.oracle.com/javase/tutorial/networking/urls/readingWriting.html>  
. Accessed 26 February 2011.
14. Java™ 2 platform standard ed. 5.0. Oracle.  
<http://download.oracle.com/javase/1.5.0/docs/api/java/lang/StringBuilder.html>.  
Accessed 28 February 2011.
15. Cadenhead R, Lemay L. Sams teach yourself Java™ 6 in 21 Days. May 2007,  
pp.457-460.
16. Sierra K, Bates B. Head first Java. O'Reilly media Inc. February 2005, pp.480.
17. Moller A, Schwartzbach M, An introduction to XML and web technologies.  
2006, pp.33-40.
18. Professional Java XML. Wrox Press Ltd., 2001, pp.54-63.

19. Deitel P, Harvey M. Deitel & Associates Inc. Java™ for programmers: Deitel® developer series. Prentice hall, February 2009, pp.1022-1024.
20. Hong. IndexOf() and lastIndexOf() in Java. 2011. <http://www.java-samples.com/showtutorial.php?tutorialid=225>. Accessed 15 March 2011.
21. Hansen M. SOA using Java web services. Prentice Hall, May 2007, pp.346-355.
22. Java™ 2 platform std. ed. v1.4.2. Oracle.  
<http://download.oracle.com/javase/1.4.2/docs/api/javax/xml/parsers/DocumentBuilderFactory.html> . Accessed 25 March 2011.
23. Hewitt E. Java soa cookbook. O'Reilly Media Inc. March 2009, pp.102-106.

## Appendices

## Appendix 1: Code for logging in

```
public LoginForm() {  
    shell.setLayout(new GridLayout(2, false));  
    shell.setText("Login form");  
    label1 = new Label(shell, SWT.NULL);  
    label1.setText("User Name: ");  
    account = new Text(shell, SWT.SINGLE | SWT.BORDER);  
    account.setText("");  
    account.setTextLimit(30);  
    label2 = new Label(shell, SWT.NULL);  
    label2.setText("Password: ");  
    password = new Text(shell, SWT.SINGLE | SWT.BORDER);  
    System.out.println(password.getEchoChar());  
    password.setEchoChar('*');  
    password.setTextLimit(30);  
    Button button = new Button(shell, SWT.PUSH);  
    button.setText("login");  
    button.addListener(SWT.Selection, new Listener() {
```

## Appendix 2: Code for URL encoding

```
String d1("&" + URLEncoder.encode("account", "UTF-8") + "=" + URLEncoder.  
encode(selected, "UTF-8");
```

```
String d2("&" + URLEncoder.encode("password", "UTF-8") + "=" + URLEncoder.  
encode(selected1, "UTF-8");
```

```
URL viope = new URL("https://vw261.viope.com/cgi-  
bin/contents/eclipse_plugin/plugin.pl?page=login"+d1+d2);
```

## Appendix 3: Code for URL connection

```
URLConnection vp = viope.openConnection();
    StringBuilder sb = new StringBuilder();
    BufferedReader in = new BufferedReader( new
    InputStreamReader(vp.getInputStream()));
    String inputLine;
    while ((inputLine = in.readLine()) != null)
    {
sb.append(inputLine).append(System.getProperty("line.separator"));
    }
}
```

## Appendix 4: Code for parsing XML file

```
// Create an instance of the DocumentBuilderFactory
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setValidating(false);
// Get the DocumentBuilder from the factory that we just got above.
DocumentBuilder db = dbf.newDocumentBuilder();
// turn it into an in-memory object
Document doc = db.parse(new InputSource(new StringReader(nohtml)));
// doc.getDocumentElement().normalize();
System.out.println("Root element " + doc.getDocumentElement().getNodeName());
NodeList nodeList = doc.getElementsByTagName("exercises");
Node fstNode = nodeList.item(0);
// Node scdNode = nodeList.item(0);
if (fstNode.getNodeType() == Node.ELEMENT_NODE) {
Element fstElmnt = (Element) fstNode;
NodeList fstNmElmntLst = fstElmnt.getElementsByTagName("exercise_template_id");
Element fstNmElmnt = (Element) fstNmElmntLst.item(0);
NodeList fstNm = fstNmElmnt.getChildNodes();
}
```

```
System.out.println("Exercise 1 ID : " + ((Node) fstNm.item(0)).getNodeValue()); }
```

Appendix 5: Code for writing XML file

```
private String configFile;  
public void setFile(String configFile) {  
    this.configFile = configFile;  
}  
  
public void saveConfig() throws Exception {  
    // Create a XMLOutputFactory  
    XMLOutputFactory outputFactory = XMLOutputFactory.newInstance();  
    // Create XMLEventWriter  
    XMLEventWriter eventWriter = outputFactory  
        .createXMLEventWriter(new FileOutputStream(configFile));  
    // Create a EventFactory  
    XMLEventFactory eventFactory = XMLEventFactory.newInstance();  
    XMLEvent end = eventFactory.createDTD("\n");  
    // Create and write Start Tag  
    StartDocument startDocument = eventFactory.createStartDocument();  
    eventWriter.add(startDocument);  
    // Create config open tag  
    StartElement configStartElement = eventFactory.createStartElement("",  
        "", "config");  
    eventWriter.add(configStartElement);  
    eventWriter.add(end);  
    // Write the different nodes  
    createNode(eventWriter, "excercise_id", my);
```

```
createNode(eventWriter, "unit", "901");
createNode(eventWriter, "current", "0");
createNode(eventWriter, "interactive", "0");
eventWriter.add(eventFactory.createEndElement("", "", "config"));
eventWriter.add(end);
eventWriter.add(eventFactory.createEndDocument());
eventWriter.close();
}

private void createNode(XMLEventWriter eventWriter, String name,
String value) throws XMLStreamException {
XMLEventFactory eventFactory = XMLEventFactory.newInstance();
XMLEvent end = eventFactory.createDTD("\n");
XMLEvent tab = eventFactory.createDTD("\t");
// Create Start node
StartElement sElement = eventFactory.createStartElement("", "", name);
eventWriter.add(tab);
eventWriter.add(sElement);
// Create Content
Characters characters = eventFactory.createCharacters(value);
eventWriter.add(characters);
// Create End node
EndElement eElement = eventFactory.createEndElement("", "", name);
eventWriter.add(eElement);
eventWriter.add(end);
}

public static void main(String[] args) {
MsOfficeAction configFile = new MsOfficeAction();
```

```
configFile.setFile("config10.xml");  
  
try {  
    configFile.saveConfig();  
} catch (Exception e) {  
    e.printStackTrace();  
}  
  
System.out.println("your file is written !!");  
}
```

#### Appendix 6: Code for Sending a file

```
String wd = System.getProperty("user.dir");  
    JFileChooser fc = new JFileChooser(wd);  
    int rc = fc.showDialog(null, "Select Data File");  
    if (rc == JFileChooser.APPROVE_OPTION)  
    {  
        File file = fc.getSelectedFile();  
        filename = file.getAbsolutePath();  
        // function is called  
        System.out.println(filename);  
    }  
    else  
        System.out.println("File chooser cancel button clicked");  
  
    File file = new File (filename);  
  
    try  
    {  
        content = FileUtils.readFileToString(file);
```

```
} catch (IOException e)
{
    e.printStackTrace();
}
content = URLEncoder.encode(content, "UTF-8");

out.write("page=test_answer&chapter_id=14608&exercise_template_id=4335&sid=4b
8f3fa4cc908567ff8587d7f699cbcd&answer="+content);
out.close();

BufferedReader in = new BufferedReader(
    new InputStreamReader(
        connection.getInputStream()));

//String decodedString;

String line; // String that holds current file line
int count = 0; // Line number of count

// Read first line
line = in.readLine();
count++;

// Read through file one line at time. Print line # and line
while (line != null){
    System.out.println(count+": "+line);
    line = in.readLine();
    count++;
}
```

```
/*while ((decodedString = in.readLine()) != null) {  
    System.out.println(decodedString);  
}*/  
in.close();  
}  
}
```

