

**Location Independent Presentations Platform**  
For Programming Lectures

**I H N Lanka**

**158764E**

Supervised by Mr.Chaman Wijesiriwardana

Master of Science in Information Technology

University of Moratuwa, Sri Lanka

February 2019

## **Location Independent Presentations Platform**

For Programming Lectures

A software tool that providing interactive and more productivity on programming sessions

**I H N Lanka**

**158764E**

Supervised by Mr.Chaman Wijesiriwardana

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the partial fulfilment of the requirements of the Degree of MSc in Information Technology.

Faculty of Information Technology

University of Moratuwa

February 2019

# Declaration

I declare that this report is my own work and has not been submitted in any form for another Masters, Degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of the Student:

Signature of the Student

I.H.N Lanka

.....

Date:

Supervised by:

Signature of the Supervisor

Mr.Chaman Wijesiriwardana

.....

Date:

## **Dedication**

This dissertation is dedicated to my beloved parents, siblings, my teachers who gave me endless courage and support to achieve my task and goal in completing the research project.

## **Acknowledgement**

My heartiest thanks go to my supervisor Mr Chaman Wijesiriwardana for the guidance, assistance, encouragement, valuable advice on improving the research and providing this opportunity to carry out this research project.

Also, sincerely thanks to all my teachers who taught in MSc IT degree program. Things learned from these subjects made it easier to make this research project a successful one.

Finally, a sincere thank goes to everyone who supported teams especially from Redot Pvt for contributing their valuable time on this research.

# Abstract

The rise of modern-day technologies also affects the classroom. When it comes to teaching and learning, they use advanced technology methods, as it's easier to communicate with the students and teacher in or out of the classrooms.

Lecture presenting becomes a significant part here, and earlier days addresses must use conventional ways to do the presentation of their speech. Starting from the projector to attractive touch screens they convert. But these methods are not enough to get interact with the students. They used to get boring and when conducting programming lectures, both students and lecture face to a primary problem, which is when they do the coding part. When the lecture shows in the presentation and gives exercises to students, it gets hard to interact with the lecture and doing the debugging. Because students have to wait till lecturer comes or lecturer should go through every student. If users need to get help with each other, they need to visit each of everyone. This part requires an external effort like the help of lab assistants.

Primary Goal of the research is to find a proper mechanism to conduct the presentation with code synchronized with lecture, student and all together. Also, there are some other methods nowadays people achieving it, like remote desktops, using an online IDE. But those have limitations.

Also, we have found a big problem that lots of people face when using FTP, as this should be port forwarded. When it comes to setup port forwarding it has many limitations like ISP blocked, router not supported and require some technical knowledge to setup. So, we have also research to find a proper solution for connecting and do FTP process without the port forwarding or any complex methods.

With this research, we can resolve all those problems and remove external efforts like lab assistants. This will affect the productivity of lecture sessions with connectivity among lecture and students. With the proposed tool, we use an online presentation platform with code syncing and debugging with every student who connected to the current session. Plus the students can raise their questions from the lecturer, without visiting and users together can help each other to solve their problems without visiting them.

Also, we can achieve the goal of connecting the File Manager process without any FTP or port forwarding methods. By installing a simple software, everyone can share their specified file location with another end. So, we connect both Presentation Platform with coding syncing to the method to achieve the File Transfer process easy.

This tool reduces the waste time of lecture and student by automating the synchronized part of coding exercise and presentations. Make more interactive and productivity classroom for both ends.

# Table of Contents

1	Introduction .....	1
1.1	Prolegomena.....	1
1.2	Background and Motivation.....	2
1.3	Problem in Brief .....	4
1.4	Aim and Objectives .....	6
1.4.1	Aim .....	6
1.4.2	Objectives .....	6
1.5	System Requirements .....	6
1.6	Structure .....	7
1.7	Summary .....	8
2	Current Development and Challenges .....	9
2.1	Introduction .....	9
2.2	Current Developments.....	9
2.2.1	A simple Presentation Tool Based on Web pages .....	9
2.2.2	Canvas Based and Slide Based Presentation Platform .....	10
2.2.3	Paranoid .....	10
2.2.4	Cloud-based IDE.....	11
2.2.5	Other works.....	12
2.2.6	Google Slides .....	12
2.2.7	Prezi .....	12
2.2.8	Ludus.....	12
2.2.9	FileZilla.....	12
2.2.10	Cyberduck .....	13
2.3	The Research Gap .....	13
2.4	Problems Identified .....	15



2.5	Summary .....	17
3	Technology foundation of the solution.....	18
3.1	Introduction .....	18
3.2	Technologies used for the solution.....	18
3.2.1	NodeJS for server-side JavaScript .....	18
3.2.2	NPM Package Manager .....	20
3.2.3	MongoDB .....	20
3.2.4	JSON .....	20
3.2.5	AngularJS for Client-Side JavaScript .....	20
3.2.6	File System Technology .....	21
3.2.7	Tunnelling .....	21
3.2.8	HTML 5 .....	21
3.2.9	jQuery .....	21
3.2.10	ExpressJS .....	21
3.2.11	Electron .....	22
3.2.12	Development Tools.....	22
3.2.13	Bitbucket .....	22
3.2.14	Visual Code.....	22
3.2.15	Ampps .....	22
3.3	Hosting and Deployment technologies .....	23
3.3.1	AWS EC2.....	23
3.3.2	MobaXtream .....	23
3.4	Summary .....	23
4	Approach .....	24
4.1	Introduction .....	24
4.2	Requirement Gathering .....	24
4.3	Issues faced by the lectures and students and in industry .....	24

4.4	Gap identified which hasn't been addressed by other researchers.....	25
4.5	Limitations with current development tools.....	26
4.6	Hypothesis.....	26
4.7	Requirement definition.....	27
4.8	Input and Output.....	27
4.9	Process.....	27
4.10	Users.....	28
4.11	Features .....	29
4.12	Summary .....	29
5	Design.....	30
5.1	Introduction .....	30
5.2	Research planning .....	30
5.3	Development methodology .....	30
5.4	Selection of the software process model.....	31
5.5	Analysis of the current development workflow .....	31
5.6	Requirement analysis .....	32
5.6.1	Functional requirements of the solution .....	32
5.6.2	Non - Functional requirements of the solution .....	32
5.7	Top Level Design.....	33
5.8	Module architecture.....	33
5.8.1	File Manager .....	33
5.8.2	Lexer .....	34
5.8.3	File Processor.....	34
5.8.4	User Interface Module .....	34
5.8.5	Backend Core.....	35
5.8.6	Database Module .....	35
5.9	Version controlling.....	36

5.10	Dependency management .....	36
5.11	Compiling and building.....	36
5.12	Testing and continuous integration .....	36
5.13	Summary .....	36
6	Implementation.....	37
6.1	Introduction .....	37
6.2	Overall solution.....	37
6.3	Implementation of the Solution.....	37
6.4	Preparation and Programming.....	38
6.5	Implementing of User Interface .....	38
6.6	Implementation of Back End.....	39
6.7	Summary .....	40
7	Evaluation.....	41
7.1	Introduction .....	41
7.2	Case study 1 – conducting interactive programming training.....	41
7.2.1	Problem Definition and Requirement Analysis .....	41
7.2.2	Problem Definition and Requirement Analysis .....	42
7.2.3	Design and Implementation .....	42
7.2.4	Evaluation .....	42
7.3	Case study 2 – Participating in an interactive webinar .....	44
7.3.1	Problem Definition and Requirement Analysis .....	44
7.3.2	Evaluation .....	44
7.4	Other Evaluations.....	45
7.4.1	Participants.....	45
7.5	Test Environment.....	54
7.6	Personal Feedback.....	54
7.7	Aim.....	55

7.8	Objectives.....	55
7.9	Drawbacks and limitations .....	55
7.10	Summary .....	55
8	Conclusion and Future Development .....	56
8.1	Introduction .....	56
8.2	Conclusion.....	57
8.3	Future work .....	57
8.4	Summary .....	58
1	Appendix A.....	61
1.1	File Project Initialization and Update to User Engine.....	61
2	Appendix B.....	63
2.1	Create Tunnel Access.....	63
2.2	Get local files and transfer to presentation.....	63
3	Appendix C .....	64
3.1	Select working directory .....	64
3.2	Enter the user logins of the system.....	64
3.3	View list of online users and projects .....	65
3.4	Presenter creating programming assessments for users .....	65
3.5	Assessments appear on the students who logged in the system.....	66
3.6	Set up the project to their pc.....	66
3.7	Users source code for programming assessment.....	67
3.8	Can see the status of the programming code with errors .....	67

# Table of Tables

Table 1 - System Requirements .....	7
Table 2 - List of Evaluation Criteria for existing prototyping tools .....	13
Table 3 - Evaluation done for existing prototyping tools .....	14
Table 4 - Evaluation Conduct in the case study 1 .....	43
Table 5 - Evaluation Conduct in the case study 2.....	44

# Table of Figures

Figure 1 - lecture without activities .....	16
Figure 2 - lecture with activity.....	16
Figure 3 - Server Creates New thread from Limited Pool or Waits for Available Thread. ....	19
Figure 4 - Handles Event-Based Call-back on Single Thread .....	19
Figure 5 - Execution plan for the proposed system .....	30
Figure 6 - Development methodology for proposed system.....	31
Figure 7 - Top level design architecture of the proposed system .....	33
Figure 8 - User Interface Module.....	34
Figure 9 - Backend Core .....	34
Figure 10 - Database Module.....	35
Figure 11 - Project Initialization Engine.....	61
Figure 12 - Project file update engine.....	62
Figure 13 - Initialize tunnel access .....	63
Figure 14 - get file and transfer to presentation.....	63
Figure 15 - File Authorizer Before Select.....	64
Figure 16 - File Authorizer after select directory .....	64
Figure 17 - View of online users.....	65
Figure 18 - creating assessment .....	65
Figure 19 - available assessment files.....	66
Figure 20 - setup project file.....	66
Figure 21 - student's code without syntax errors .....	67
Figure 22 - code with error and error listing.....	67