22pp.pdf

*by*

**Submission date:** 11-May-2020 04:27PM (UTC+0530)

**Submission ID:** 1321595845 **File name:** 22pp.pdf (1.22M) **Word count:** 3619

**Character count:** 21442



###### A Project Report

On

*PLAGIARISM CHECKER*

*Submitted for the partial fulfillmentof the req*f4 *irement for the award of the Degree of*

*B.Tech*

*Computer Science and Engineeri*n*g*

by

*Harshit Agrawal f170102314)*

*Prathmesh Jain {180102906)*

*Utkarsh Saxena {170102174)*

*Satyam Bhatia {170102167)*

Under the Guidance of

*MrXushal* Cr *upta*

Asst. Professor, DIT Univeristy, Dehradun

* I a ' I I " k H r' ' r• '''

DIT UNIVERSITY, DEHRADUN, INDIA

April 2020

DECLARATION

This is to certify that the Project entitled “Plagiarism Checker”in partial fulfillment of the requirement for the award of the Degree of Bachelor of Technology in Computer Science and **Engineering** , submitted to DIT University, Dehradun, Uttarakhand, India, is an authentic record of bonafide work carried out by me, under the guidance of Mr. Kushal Gupta (Asst. Prof. D lTU).

The matter emhndied in this Project/ Thesis/ Dissertation has not been submitted for the award of any other degree or diploma to any University/ Institution.

Harshit Agrawal Utkarsh Saxena Prathmesh Jain Satyam Bhatia

Students Name **& Signature:**



Dr. Vishal Bharti **Head** of **Department Signature**

*Date:24 A pril2G20 Pface: Dehradun*

Mr. Kushal L’upta

(Asst. **Prof. DITU,Dehradun)**

Supervisor Name , Designation

& Signature:

### ACKNOWLEDGMENT

\* We are grateful for the golden opportunity that I got in DIT University which helped us to improve and learn things which helped us to improve and learn things for the better. Hence, we find ourselves extremely fortunate to be given a chance of being a part of such an event. We are obliged that during this phase we met such helping Faculty along with superb individuals who helped us throughout the phase.

We are immensely thankful to Mr. Kushal Gupta, our module lead who helped us by providing his valuable ideas along with his precious advices. His presence made everything simpler and easy for us.

We look at this opportunity as a integral advancement in our professional experience. We will make sure that we utilize the most of our experience and learnings for our future betterment and we will keep on trying to upgrade ourselves in the coming time so as to achieve the specific goal.

.

.

Abstract

For detecting the plagiarism of any kind, it is important to know about its all possible forms and cases, and about the existence of different variety of tools and systems which are used for its detection. According to the impact and severity of harm, plagiarism may exist in an article or in any major production of project in a number of ways. These surveys present taxonomy of multiple plagiarism forms and involve discussion these different plagiarism forms. Over recent years, many tools and techniques have come into existence to detect all forms of plagiarism.

IV



TABLE OF CONTENTS

Title Page No.

ACKNOW LEDGEMENT. iii

ABSTRACT. iv

TAB LE OF CONTENTS. v

LIST OF FIG URES. vii

AB BRL VIA TIONS. .. .. .. . . . . . . . .... . . . . . . . .. .. .. .. .. . . .. .. . . .. .. .. . ..

CHAPTER 1 INTRODUCTION

1.1 Introduction ......................... ............................... . . .....

CHAPTER 2 Progress

2.I Progress...... ... ...... . . . . .. ........ ...... .................. ........ . . .. ....

CHAPTER 3 C.allery

3. I Images............................. ............................. ............

CfIAPTER 4 Technolojpes Used

4.1 Node J S. . . . . .. . . . . .. . . . . . . . .. . . .... . . . . . . . .. . . . . . . . .. . . . . . . . .. . . ..................

4.2 Express. . .. . . . . . . . .. . . . . . . . .. . . .. . .. . . . . . . . .. . . . . . . . .. . . . . . . . .. . . . . . ..... . . . ..

4.3 M On g OD k . . . . . . . . . .... ... ... ... ... ... . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . .

4.4 Bootstrap. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ...

4.5 HTML.. . . .. .. .. . . . . . . . . . . . . . . . .. .. . . .. .. .. .. .. . . . . . . . . . . . . . .. . . .. .. .. .. .. ....

4.6 Javascript ............... ............. . ............. ................ .. ... .....

4.7 CSS. .. .. . . .. .. . . . . . . . . . . . .. .. . . .. .. . . .. .. .. . . . . . . . . . . . . . . . .. .. . . .. .. .. .. .. . ...

CHAPTER 5 Code

5.1 User Model. . .. . .. .. . . .. ................. . . .. . . . . .. . ... .. .. .. .. .. .. .. ... . . ..

5.2 Server File. . .. . . . . .. . ... .. .. .. .. .. .. .. ... . . . . . . . ... .. .. .. .. .. .. .. ..... ........

CHAPTER 6 Future Scope

6.1 Future Scope. . .. .. .. .. .. .. .. ............. . . . . .. .. .. .. .. ............. . .......

CHAPTER 7 Plagiarism DetecDon Methods

7.1 Plagiarism detection Methods.. .. . . . . . .. .... . . .. . .. .. . . .. .. ...............

7.2 Similarity Percentage Calculation.. .. .. .. .. ............. . . . . .. .. .. .. ...

REFERENCES

VI

## LIST OF FIGURES

###### Figure No. Title Page No.

* 1. Home Page. x
  2. sign Up Page. ,z
  3. [Login Page. xi](#_TOC_250005)
  4. [Dashboard Page. xi](#_TOC_250004)
  5. [Updates page. xii](#_TOC_250003)
  6. HOUSS€ S £tQe XII
  7. [Activity Page. xiii](#_TOC_250002)
  8. [Submission Page. xiii](#_TOC_250001)
  9. [Profile Page. xiv](#_TOC_250000)

VI I

ABBREVIATIONS

API = Application Program Interface HTTP = H yperText Tras ter Protoc ol HTML = HyperText Markup Language

CSS = Cascading Style Sheet

SQL = Structured Query Language

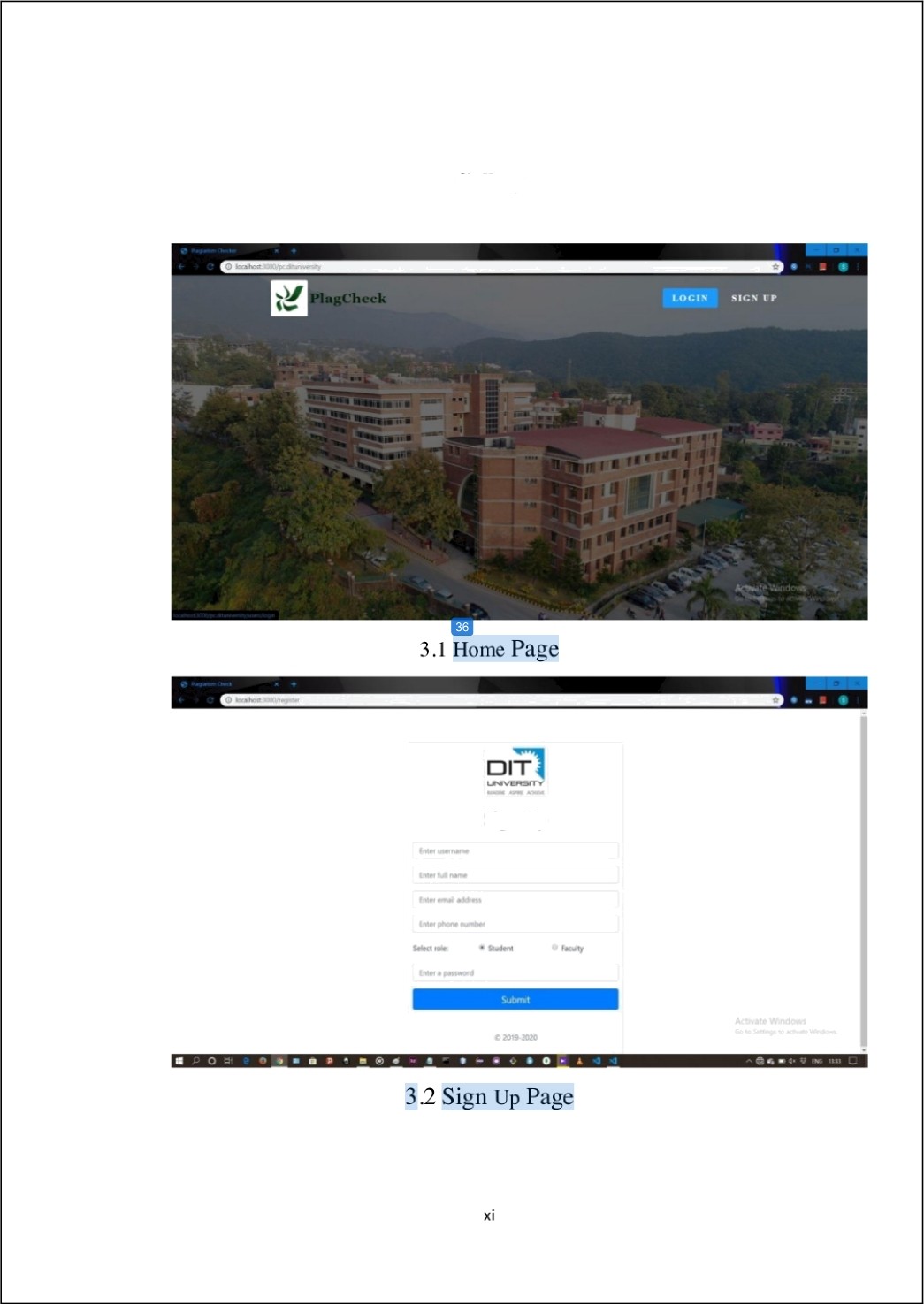
**INTRODUCTION**

In these modern times , the digital resources in the World Wide Web have been increasing gradually . This growth has also resulted in the growing chances of duplicacy and violation of copyrights . Researchers and programmers have been working continuously to check this problem of duplicacy which is known as Plagiarism.

Plagiarism is referred to as copying of someone else' lines , data , idea , vison and presenting it as one's own . Basically plagiarism is copying someone's ideas , work and words and then presenting all these without the acknowledgement of the actual source of these ideas , works and words . There are mainly 2 kinds of plagiarism which are mostly found in real world, viz " plagiarism in text " and " plagiarism in source code " . A number of programmers and software teams have been consistently trying to provide the generation with an efficient way of detecting plagiarism.

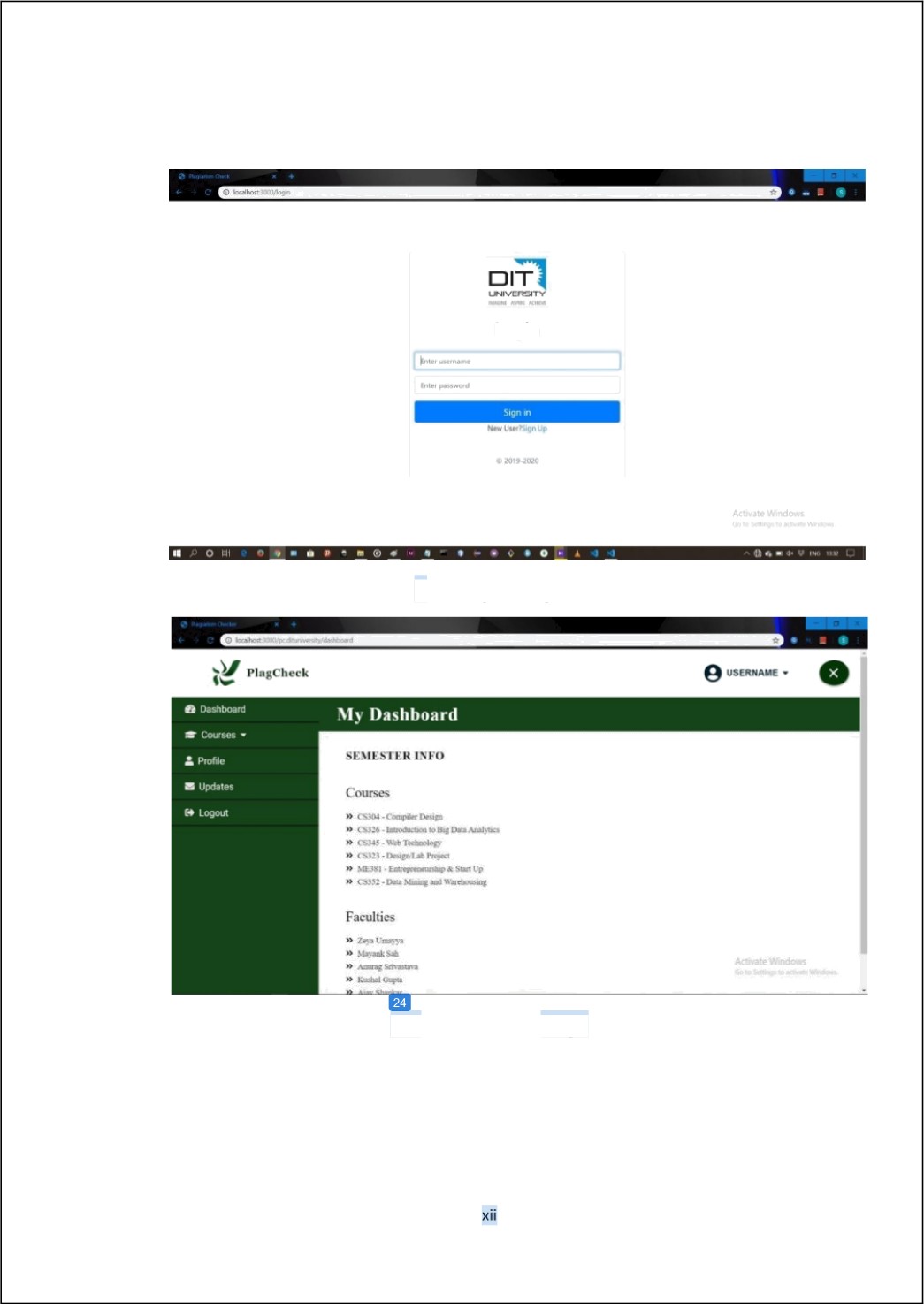
#### Progress

Till now we hav’e worked on the front end part of our project. We have developed a landing page i.e,when we go to site the landing page will be shown which will contain the 3 options- about, sign u p, sign in. Our landing page is shown in the below picture. lf we alread y have an account on this site we can directly log in to our account and find the option to u pload our assignment which can be checked for plagiarism else we need to create our account using the sign u p option available at right corner of the page. While signing u p for new account it will ask your role as in student signup or the facult y sign up, we need to choose any one according to our role. Once we are successfully logged in we are taken to the page where we can upload our file for plagiarism detection.



Gallery

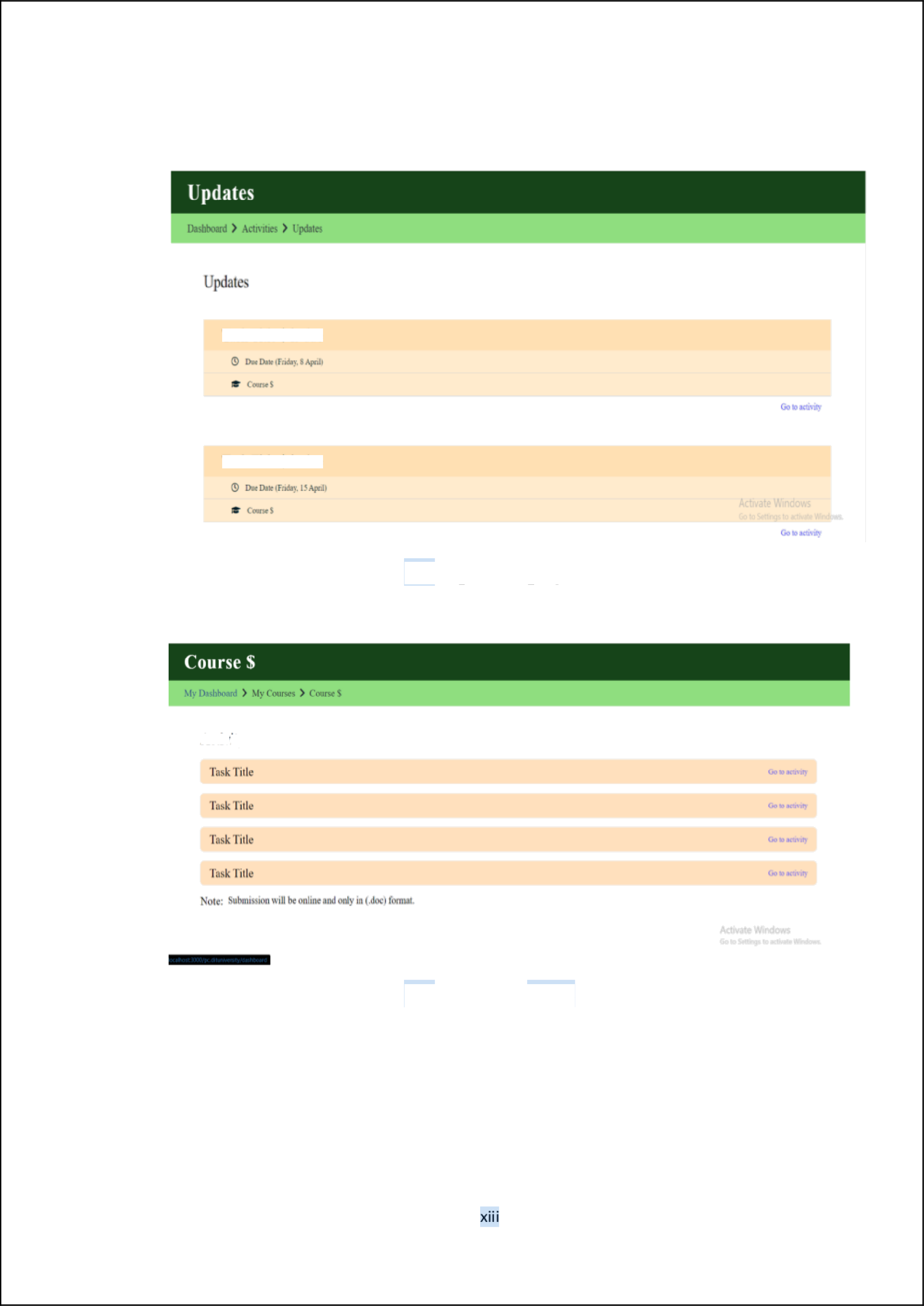
Sign Up



Login

## Login Page

## Dashboard Page

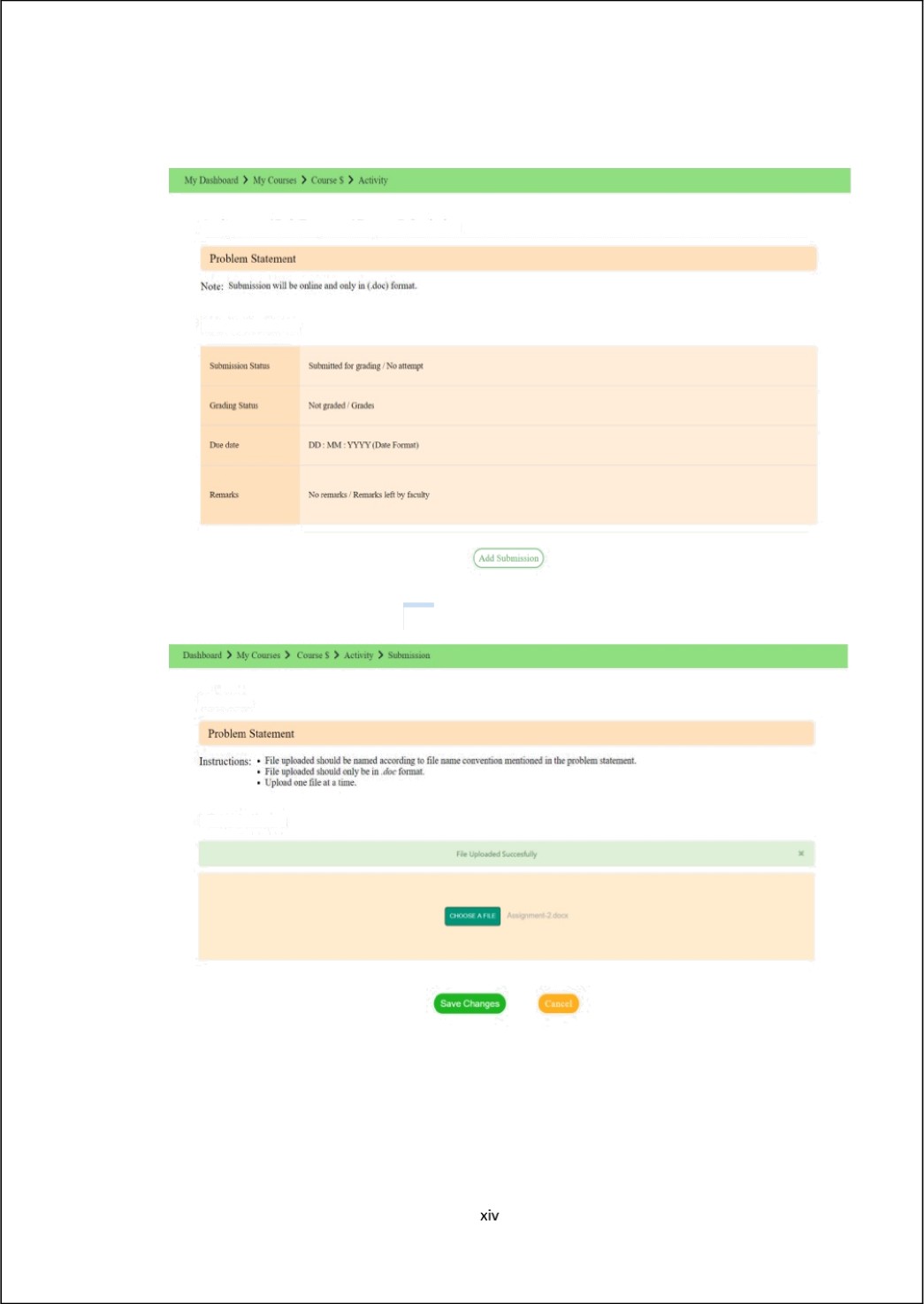


Task Title $ is due

Task Title $ is due

## Updates page

* 1. Courses Page

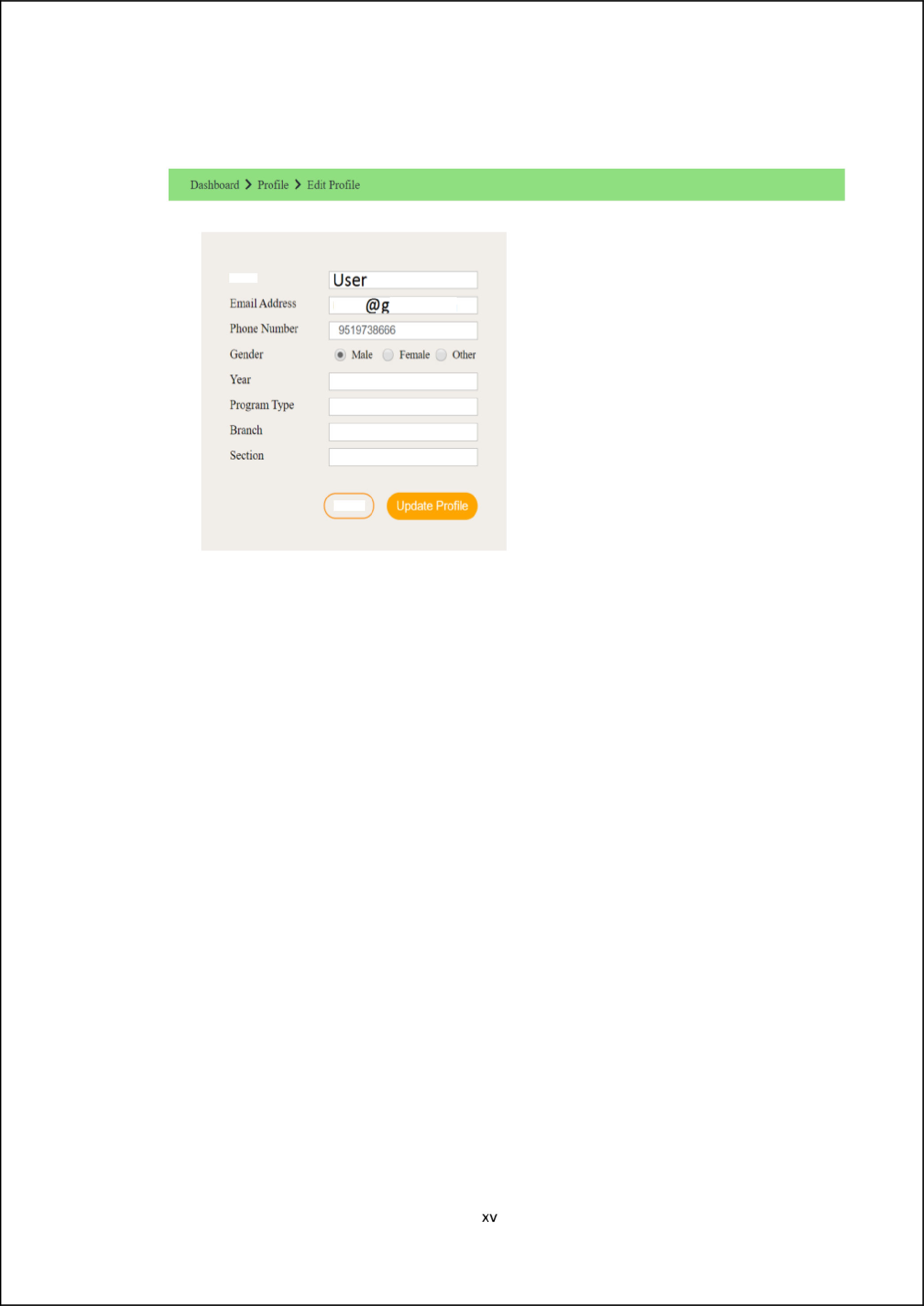


Assignment / Lab Program / Repoñ Submission

## Activity Page

Task Title

## Submission Page



It

user mail.com

## Profile Page

Technologies Used

Node JS (Language):

As an asynchronous event-driven Jag,jScript runtime work, Node.js is designed to build scalable network application .lt is similar in design , and influenced by systems like Ruby's Event Machine and Python's Twisted. Node.js takes the event model a little further. lt presents an event loop as a runtime construct instead as a library. In other systems, there is always a blocking call to start the event-l‹xip. Typically behavior is defined through callbacks at the beginning of a script & at the end a server is started through a blocking call like EventMachine: :run(). Here, there is no such start-the-event-loop call. Ncde.js simply enters the event loop afier executing the input script. lt exits the event loop when there are no more callbacks to perform. This behavior is like browser JavaScript, the event loop is hidden from the user.

**HTTP** is a first-class citizen in Node.js, designed with streaming and low latency in mind. This makes Ncde.js well suited for the foundations of a web library or framework. N‹xte.js, which is being designed without threads doesn't mean we can't take advantage of multiple cores in its environment. Child processes can be spawned by using our child process.fork() API, and is designed to be easy to communicate with. Built upon that same interface is the cluster module, which allows you to share sockets between processes to enable load balancing over your cores.

Express (Server Side):

Express is a web application framework for Node.js that allows us to spin up robust

API and web server in a much easier and S aner ways. Express is a lightweight package that does not obscure any of the core Node.js features.

The author, TJ Holowaychuk, described express as a Sinatra-inspired server I citation needed j, meaning that it is relativel y minimal with many features available as pl ugins. Express is the back-end component of the MEAN stack, together with the MongoDB database software and AngularJS front-end framework.

MongoDB (Database) :

MongoDB is a cross-platform document-oriented database program. MongoDB is Classified as a NoSQL database program, which uses JSON-like documents with schema. MongoDB is developed by MongoDB Inc. and licensed under the SSPL.

MongoDB supports field, range query, and regular-expression searches. Queries can return specific fields of documents and also include multiple user- defined Javascript functions. These Queries can also be configured to return a random sample of results of a given size.Fields in a MongoDB document can be indexed with primary and secondary indices.

lt provides high availability with replica set. A replica set consists of two or more copies of some data. Each replica-set member acts in the role of primary or secondary replica at any time. All writes and reads are done on the primary replica by default. Secondary replicas maintains a copy of the data of primary using built- in replication. When the primary replica fails, the replica set automatically conducts an election process to determine which secondary should become the primary. Seconder y can optionally serve read operations, but that data is only eventually consistent by default.



Bootstrap is an open source toolkit for developing with the help of **HTML,** CSS, and JS. lt quickly prototype ideas or builds entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plug-in built on jQuery.

**Npm libraries/dependencies** used:

* + 1. body-parser
    2. express
    3. express-fileupload
    4. express-session

1. passpon
2. passpon-local passport-local- mongoose

HTML:

HTML is the standard markup language for documents to be displayed in a web browser. LT can be assisted with technologies such as CSS and scripting languages ​​such as JavaScript. Web browsers receive HTML documents from a web service or from local storage and move this data into multimedia web pages. HTML defines a web page layout and the source y entails the origin of the documents.

HTML elements are the building blocks of HTML pages. With HTML developers, images and other things like functional forms can be imported into a translated page. HTML provides a way to create structured documents by displaying structured semantics of text such as topics, categories, lists, links, quotes and other things. HTML elements are defined by tags, written using angle brackets.

Tags like "<img> and <input /> import directly into the content of the page and tags like <p> surround and provide information about document text and possibly include other tags as sub-items. Browsers do not display HTM L tags, but use them to interpret the page content.

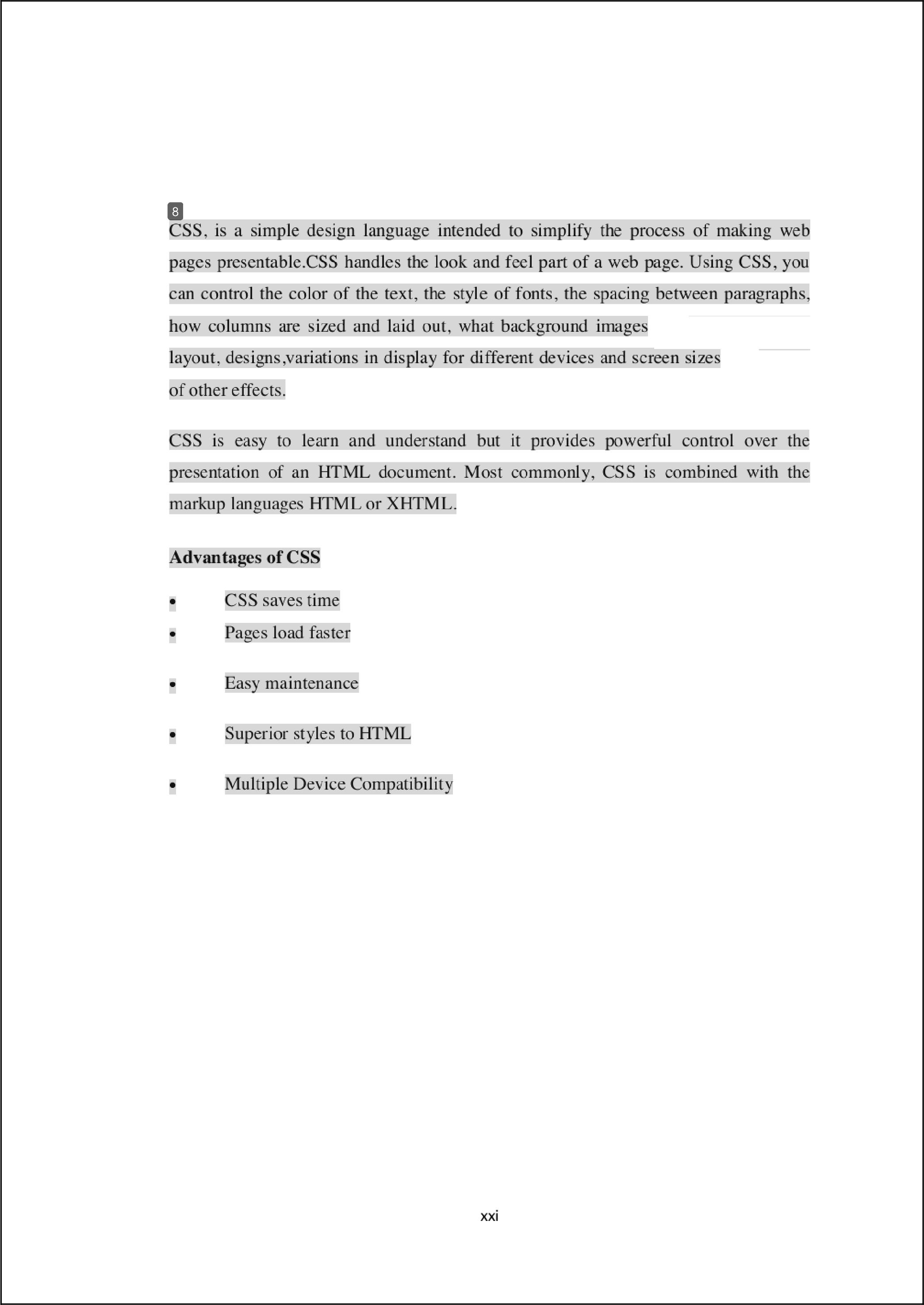
HTML can embed programs written in scripting language such as avascript, which affect the functionality and content of web pages. CSS installation defines the appearance and structure of content. The World Wide Web Consortium (W3C), a former rnaintainer of HTML and currently maintained by CSS standards, has promoted the use of CSS with formal HTML visible since 1997.

##### JavaScript:

###### JavaScript is a lightweight, interpreted, object-oriented language with first-class functions. lt is best known as the scripting language for Web pages, but it is used in many non-browser environments also. lt is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming st yle.

JavaScript urns on the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event. JavaScript is an easy to learn and also powerful scripting language, widely used for controlling web page behavior.Contrary to popular misconception, JavaScript is n‹›i "Interpreted Java". In a nutshell, JavaScript is a dynamic scripting language supporting prototype based object construction. The basic syntax is intentionally similar to both Java and C++ to reduce the number of new concepts required to learn the language. Language constructs, such as if statements, for and while lr›nps, and switch and try catch blocks function the same as in these languages (or nearly so).

lt functions as both a procedural and an object oriented language. Objects are created programrnatically in J avascript, by attaching methods and properties to otherwise empty objects at run time, as opposed to the syntactic class definitions common in compiled languages like C++ and Java. Once an object is constructed it can be used as a blueprint for creating similar objects.



CSS:

and colors arised,

and variety

CODE

User Model:

var rnongoose = require( "mongoose"),

passportLncalMongoose = require("passport-local-mon goose");

//SCHEMA SETkI P

var userschema = new mongcinse.Schema(( username: String,

name: String, email: String, phone: String, role: String, password: String

userscherna. plugin(passportkncalMongoose) ; module.exports = mongoose.m‹xlel(" User", user schema) ;

rver File:

var express = require( "express"), app = express(),

b‹xlyParser = require( " txi dy-parser"), mongoose = require("rnongoose"), passport = require("passjxirt"),

LocalStrategy = require( "passport-local"), passportLocalMongoose = require("passport-local-inongoose"), upload = require('express-fileupload'),

User = require("7models/user™), port = 3000;

//Application Configuration mongoose.connect("mongodb://localhost/DITUniversity™); app.set("view engine " , "ejs") ; app.use(bodyParser.urlencoded({ extended: true })); app.use(express.static dirname + "/public ")); app.use(require("express-session")((

secret: "Plagiarism Checker is running.",

resave: false, saveUninitialized: false

## })):

app.use(passport.initialize()) :

app.use(passport.session());

passport.use(new LocalStrategy(User.authenticate())); passport.serializeUser(User.serializeUser()); passport.deserializeUser(User.deserializeUser());



app.use(u pload());

// app.use(function(request, response, next) (

// response.locals.current User = request.user;

//----RO kITES-- --

app.get("/" , function(request, response) (

response.render("landing");

app.get("/horne", isMggedln, function(request, response) { response.render("files/home", ( error: "", filename: " " , message: "" });

//Authentication Routes

.get("/register", function(request, response) (

response.render("files/register");

// app.post("/register", function(request, response) (

// User.register(new User(request.body.user), request.bed y.password, function(error, user) (

*ff* if (error) (

*ff* console.log(error);



// //alert("Something went wrong. Please try again!"); response.redirect("/register");

// passport.authenticate("local")(request, response, function() ( response.redirect("/home");

app.}x›st("/register", function(request, response, next) (

User.register(new User(request.body.user), request.body.password, function(error, user) (

if (error) (

console.log(error);

//alert( "Something went wrong. Please try again!"); response.redirect("/register");

next();

}, passport.authenticate("local", { successRedirect: "/home",



failureRedirect: "fhome "

## 1))

app.get("/login", function(request, response) ( response.render("files/login ");

app.post("/login", passport.authenticate( "local " , ( sucoessRedirect: "fhome™ ,

failureRedirect: "/login"

}), function(request, response) (});

app.get("/logout", function(request, response) { request.logout();

response.redirect("/");

app.post("/upload", function(request, response) { if (request.files) (

var file = request.files.upfile,

AXVI

name = file.name, type = file.miinetype;

var uploadpath = dirname + '/uploads/' + name;

file.ink’(u ploadpath, function(err) ( if (err) (

console.log("File U pload Failed", name, err);

response.render("files/home", ( error: " true", filename: "" , message:

"Something went wrong. Please try again!" });

} else (

response.render("files/home", { error: false, filename: name, message: "

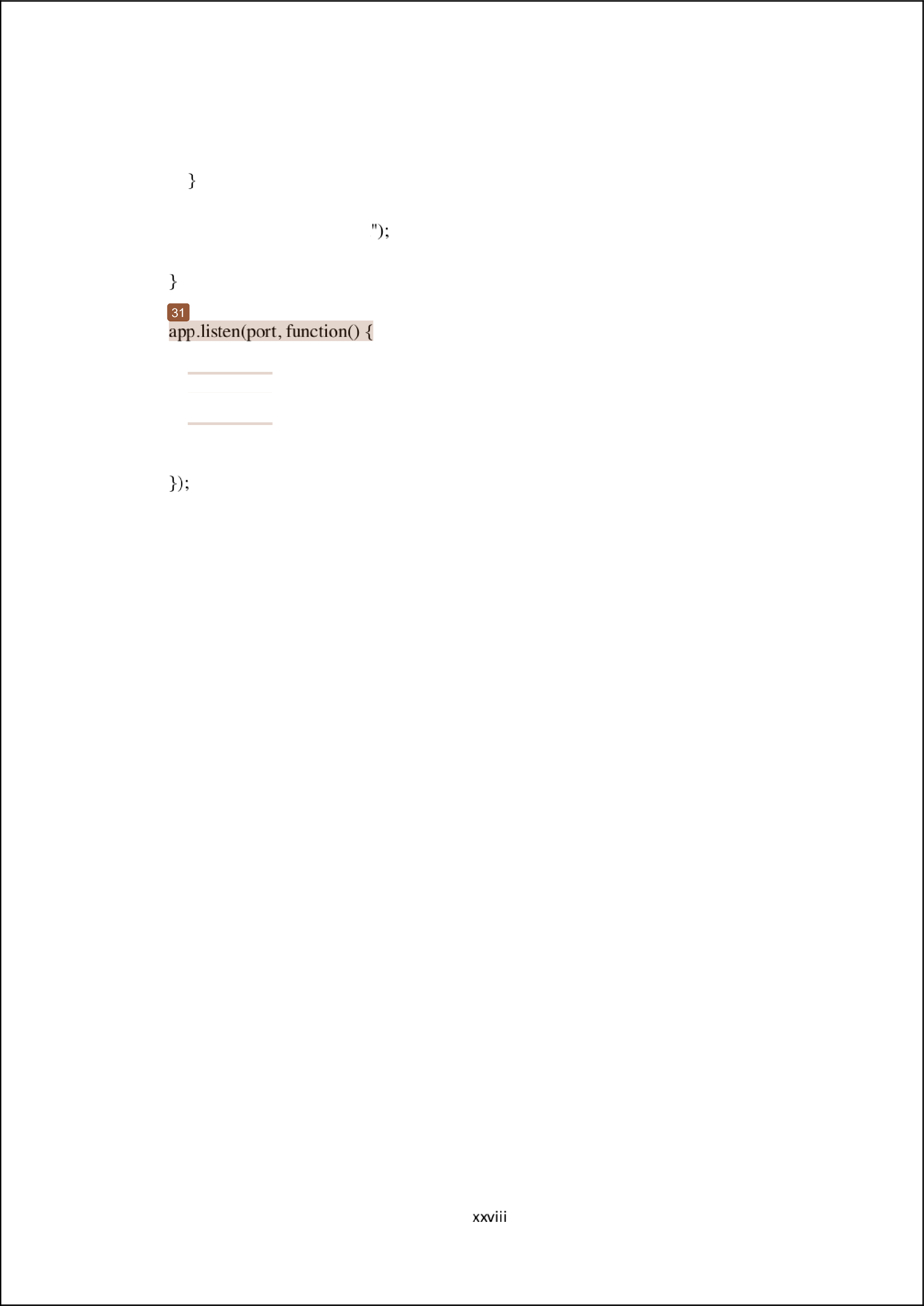
u ploaded Successfull y!" });

} else (

response.redirect("files/home", ( error: "true", filename: "", message: "No file selected ! " });

function isLogged1n(request, response, next) ( if (request.is Authenticated()) (

return next();



response.redirect("/login

console.log("Server is running on port " + }xirt); console.log("Press CTRL + C to stop the server.");

**Future Scope**

After starting the server, landing page is loaded from where only 2 requests can be made. One to the login page, second to the sign-u p page for new user,.We see 2 more requests in future about page showing information about DIT Universit y and last to the about page showing info about the project.During registration of a new user, user is asked about his details and it is also marked whether the user is faculty or a student. Then, the user logs in b y his username and password.

After login, homepage appears based on the role of user, different page for facult y and different for students.Where students are onl y allowed to see and edit their profiles and u pload the assignment, report files to their respective facult y while faculties have have the authorit y to see all the files uploaded by the students and operate on those files.Operations like checking files for plagiarism and generating results, remarks for each su bmission, grading each student. Later st udents can check these grades or remarks made b y the faculties on their submission.

**Plagiarism** Detection Methods

###### Detection of plagiarism in text document with high accuracy is a challenging task. In the past two decades, a large number of meth‹xls have been reJxirted by researchers to handle this task. These methods can be classified into eleven distinct categories. Some prominent methods under each of these categories are discussed next.

1. **Character-Based Methods:** Mostly plagiarism detection method belongs to this category. These methods exploit character-based, word-based, and syntax -based features. It utilizes these features to find similarity between a query document and existing documents. However, the similarity between a pair of documents is estimated using both exact matching and approximate matching. In exact matching, every letter in both the strings must be matched in the same order. Our survey reveals that most detection techniques are developed bled on n-gram or word n-gram based exact string similarity finding approach. F use character 16-gram matching, whereas the authors of use word 8-gram matching. Similarly, some researcher has made an effective use of approximate string matching approach. This string matching shows degree of similarity/dissirnilarity between two strings.
2. **Vector-Based Method:** Here, lexical & syntax features are extracted and categorized as tokens instead of strings. The similarity can be computed using various vector similarity measures like J accard, Dice's, Cosine, Euclidean and Manhattan coefficients. Our observation is Cosine coefficient and Jaccard coefficients are popular and effective in finding similarity between two vectors. Cosine coefficient in detecting partial plagiarism without sharing documents content. Hence it is s’ery useful to detect plagiarism in documents where submissions are considered as confidential.
3. **Syntax-Based Methods:** These methods exploit syntactical features like part of speech of phrase and words in different statements to detect plagiarism. The elements of basic POS tag are verbs, nouns, pronouns, adverbs, prepositions, conjunctions and interjections. ln , the authors use POS tag features followed by string similarity metric

to analyse and calculate similarity between texts. The authors of use syntactical POS tag to represent a text structure as a basis for further comparison and analysis.

1. **SemanDc-Based** Methods: A sentence may be defined as an ordered grouJof words. Two sentences may be same but the order of their words may be different. The degree of similarity between two words used in knowledge-based measures by Gelbukh is calculated using information from a dictionary. This similarity between two words is used as semantic similarity between two words. In another approach, Resnik used WordNet to calculate the semantic similarity, whereas, Peacock's et al., determine semantic similarity by counting the number of nodes of shortest paths between two concepts.
2. **Structure-Based Methods:**n Unlike those methods above, developed based on

lexical, syntactic, and semantic features of the text in documents to find similarity between two documents, a structure based method uses contextual similarity such as how the words are used in tire documents. However, our survey can find a few methods of this category. In the author detects plagiarism in 2 steps. I " step performs document clustering and candidate retrieval using tree-structure feature representation

and 2" step detects by utilizing M L-SOM.

n

**fi. Methods for Cross-Lingual Plagiarism DetecDon:** Cross-lingual plagiarism

detection is a very challenging task. lt require in depth knowledge of multiple languages. Finding appropriate similarity metric for such meth‹xl is also an important issue. This type of methods work based on cross-lingual text features. Various types of these methods incluJ cross-lingual syntax based methods and cross-lingual dictionary based method. A statistical model is used to evaluate the similarity between two documents regardless of the order in which the terms appear in suspected and original documents.

**Similarity Percentage Calculation**

Siinilarit y percentage is t he comparation degree of percentage similarit y between two tested documents. This similarit y will give a result of a score which will be a reference for determining percentage similarit y degree on a these docu ments. The nu mber of similar it y percentage is affected by the similarit y degree from these documents. lf similarit y percentage is higher, then simil arit y degree will be higher . Equation for calculating similarit y percentage by dividing number of similar sentences detected with total sentences in document. There are three ways for determining similarit y between docu ments:

* + Testing results lower than 30°/c means these docu ment considered hav’e little plagiarism.
  + Testing results between 30-70°/c means these docu ment considered have moderate plagiarism.
  + Testing results more than 70\*/c means these docu ment considered have heavy plagiarism.

Ref’erences

* [https://www.plagiarism.org/article/what—is—plagiarism](http://www.plagiarism.org/article/what)
* ht Fs: ncvonFrojccts.com/online-assignment-plagiarism-c hecker-project-using-data- rnining/
* https://clysiumpro.in/plagiarisms hcc ker-tool/

|  |  |  |  |
| --- | --- | --- | --- |
| 22pp.pdf |  | | |
| ORIGINALITY REPORT |  |  |  |
| 71% | 63% | 19% | 50% |
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |
| PRIMARY SOURCES |  |  |  |

# arxiv.org

1

Internet Source

[www.filemakr.com](http://www.filemakr.com/)

2

Internet Source

developer.mozilla.org

3

Internet Source

nodejs.org

4

Internet Source

en.wikipedia.org

5

Internet Source

Submitted to Amity University

6

Student Paper

Submitted to Symbiosis International University

7

Student Paper

easyanimation.in

8

Internet Source

Submitted to RK University

9

Student Paper

26%

6%

6%

6%

5%

3%

3%

2%

1%

# Submitted to University of Ulster

10

Student Paper

[www.digitalocean.com](http://www.digitalocean.com/)

11

Internet Source

Submitted to National College of Ireland

12

Student Paper

Submitted to University of Newcastle upon Tyne

13

Student Paper

Submitted to Birkbeck College

14

Student Paper

[www.dituniversity.edu.in](http://www.dituniversity.edu.in/)

15

Internet Source

Submitted to Higher Education Commission Pakistan

16

Student Paper

blog.pcollaog.cl

17

Internet Source

Submitted to National University of Ireland, Galway

18

Student Paper

Submitted to University of Hertfordshire

19

Student Paper

Azat Mardan. "Pro Express.js", Springer Science and Business Media LLC, 2014

20

Publication

1%

1%

1%

1%

1%

1%

1%

<1%

<1%

<1%

<1%

# gitlab.doc.ic.ac.uk

21

Internet Source

peaceplantsandpushups.com

22

Internet Source

asknodejs.blogspot.com

23

Internet Source

Submitted to Thammasat University

24

Student Paper

publicationslist.org

25

Internet Source

Submitted to Metropolia Ammattikorkeakoulu

26

Student Paper

[www.hackerstribe.com](http://www.hackerstribe.com/)

27

Internet Source

Submitted to Sri Lanka Institute of Information Technology

28

Student Paper

Submitted to Middle East Technical University

29

Student Paper

baadalsg.inflibnet.ac.in

30

Internet Source

gitlab.ecs.vuw.ac.nz

31

Internet Source

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

# [www.djamware.com](http://www.djamware.com/)

32

Internet Source

Submitted to RMIT University

33

Student Paper

dspace.cusat.ac.in

34

Internet Source

Practical Node js, 2014.

35

Publication

Submitted to Manchester Metropolitan University

36

Student Paper

Submitted to VHS Virtual High School

37

Student Paper

<1%

<1%

<1%

<1%

<1%

<1%

###### Exclude quotes On Exclude bibliography On

Exclude matches < 5 words