



## **15CSE376 – NET CENTRIC PROGRAMMING**

**B. Tech - VII Semester - IV Year - CSE - E**

### **PROJECT REPORT**

**GROUP - 6**

**Title: Paper Publication Tracking Portal**

<b>Roll Number</b>	<b>Name</b>
CB.EN.U4CSE18410	Arvind T
CB.EN.U4CSE18426	Jeev S S
CB.EN.U4CSE18452	Sandhya S
CB.EN.U4CSE18457	Sivasini Netra S A
CB.EN.U4CSE18469	Sidharth R

# Table of contents

S No.	Title	Page No.
1	Introduction	
2	Modules 2.1 Login & Register 2.2 Author Dashboard 2.3 Editor Dashboard 2.4 Reviewer Dashboard 2.5 Settings 2.6 Publication Search 2.7 Administrative Panel	
3	Technologies Learnt 3.1 Spring Boot 3.2 JSP 3.3 HTML 3.4 CSS 3.5 Bootstrap 3.6 Java Script	
4	System Requirement Specification 4.1 Hardware 4.2 Software	
5	Schema Diagram	
6	Architecture Diagram	
7	Database Description 7.1 Paper 7.2 Published 7.3 Inprogress	

	<b>7.4</b> Paper_published <b>7.5</b> Editor <b>7.6</b> Student <b>7.7</b> Login <b>7.8</b> Reviewer <b>7.9</b> Comments	
<b>8</b>	Frontend Screen Snips	
<b>9</b>	Servelets and Database Connectivity	

## 1. Introduction

Every year there are lots of research papers published, with the increase in domains the number papers published are also increased, so to keep track of the paper published by every author we aim to develop a Publication Tracking Portal.

A Publication Tracker is a portal that tracks papers made by students and researchers alike. It tracks the status of their paper and lets them know whether they have been published and where they have been published.

The portal starts with the Author uploading their manuscripts and an editor will be assigned to the paper published by the author who verifies whether the Basic Requirements met and then assign reviewers. After Reviewers review and give recommendations, the editor collects the reviews and decides whether to Accept, Reject or ask the author to revise the paper and go through the process again.

In our project we are planning to add SMS functionality to notify the Author about the status of the paper. We are using the latest tech to implement our project. So, it can be adapted, developed and improvised further in the future.

## 2. Modules

### 2.1. Login & Register

- There are three users who use the publication tracking portal. They are author, editor and reviewer. All the three users can log in through single login system.
- There is a separate registration page as well for new users.

### 2.2. Author Dashboard

- Author has the capability to submit their manuscript through online.
- A e-form is available wherein the author can fill the required details along with the copy of the manuscript.
- A dashboard is provided where they would be able to view the status of their paper.

### 2.3. Editor Dashboard

- The submitted manuscript by the author goes to the respective selected journal/conference database.
- There are 5 stages of reviewing a paper and each stage is updated by the editor.  
They are  
Stage 1 – To see if basic requirements are met  
Stage 2 – Allot reviewers who are of the same domain as that of the paper  
Stage 3 – Reviewers review the paper and the comments are forwarded to the editor  
Stage 4 – Based on the comments, the editor makes a decision  
Stage 5 – The editor chooses one among the following which is Reject, Accept or request for revision by author.

## 2.4.Reviewer Dashboard

- The reviewer's job is to review papers allotted by the editor and forward their comments back to the editor.

## 2.5. Settings

- Each user will be able to edit their profiles, change settings which include themes, password etc.

## 2.6. Publication Search

- All the users have the facility to browse through others publications with additional facilities provided by the site such as sorting and filtering system.

## 2.7.Administrative Panel

- This is where the site administrator takes care of the database and has the capability to apply modifications on it as well.

# 3. Technologies Learnt

## 3.1. SPRING BOOT

Spring Boot is an open-source Java-based framework used to create micro services. The micro services architecture allows developers to develop and deploy services independently. The independence translates to each service having its own process and helping achieve a lightweight model.

A few advantages of micro services are:

1. Easy deployment
2. Simple scalability
3. Compatibility with containers
4. Minimum configuration
5. Lesser Production time

Spring Boot Auto Configuration automatically configures a Spring application based on the JAR dependencies that are added in the project. The `@EnableAutoConfiguration` annotation is used for this purpose in the main class file. The entry point of the application is the class that contains the `@SpringBootApplication` annotation, this annotation negates the need for the Auto Configuration annotation. The application entry point also removes the need to NCP Project Documentation 2 include the `@ComponentScan` annotation, which scans for all the package declarations when the application initializes.

Spring Boot applications can be built using both Gradle and Maven. Our project uses Maven. The "pom.xml" file is the build file that defines the dependencies, plugins and properties. With the introduction of Maven wrapper, it is possible to run a Maven project even without having Maven installed in a PATH and it also automates the downloading of the correct version of Maven required by the project. The wrapper creates two files

“mvnw” (Linux Environment) and “mvnw.cmd” (Windows environment) which enable the automation of builds in their respective environments. A few packages used that enable http connectivity and authentication with session management:

- `Http.ResponseEntity`
- `Security.authentication.AuthenticationManager`
- `Security.crypto.password.PasswordEncoder`
- `Web.bind.annotation.RestController`
- `web.bind.annotation.RequestMapping`

### 3.2.JSP

**JSP** technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.

A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

The JSP pages follow these phases:

- Translation of JSP Page
- Compilation of JSP Page
- Classloading (the classloader loads class file)
- Instantiation (Object of the Generated Servlet is created).
- Initialization ( the container invokes `jspInit()` method).
- Request processing ( the container invokes `_jspService()` method).
- Destroy ( the container invokes `jspDestroy()` method).

### 3.3.HTML

**HTML** (Hypertext Markup Language) is the code that is used to structure a web page and its content. HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

### 3.4.CSS

**CSS** is the language used for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers.

### 3.5.Bootstrap

**Bootstrap** is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

### **3.6.JavaScript**

JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

## **4. System Requirement Specifications**

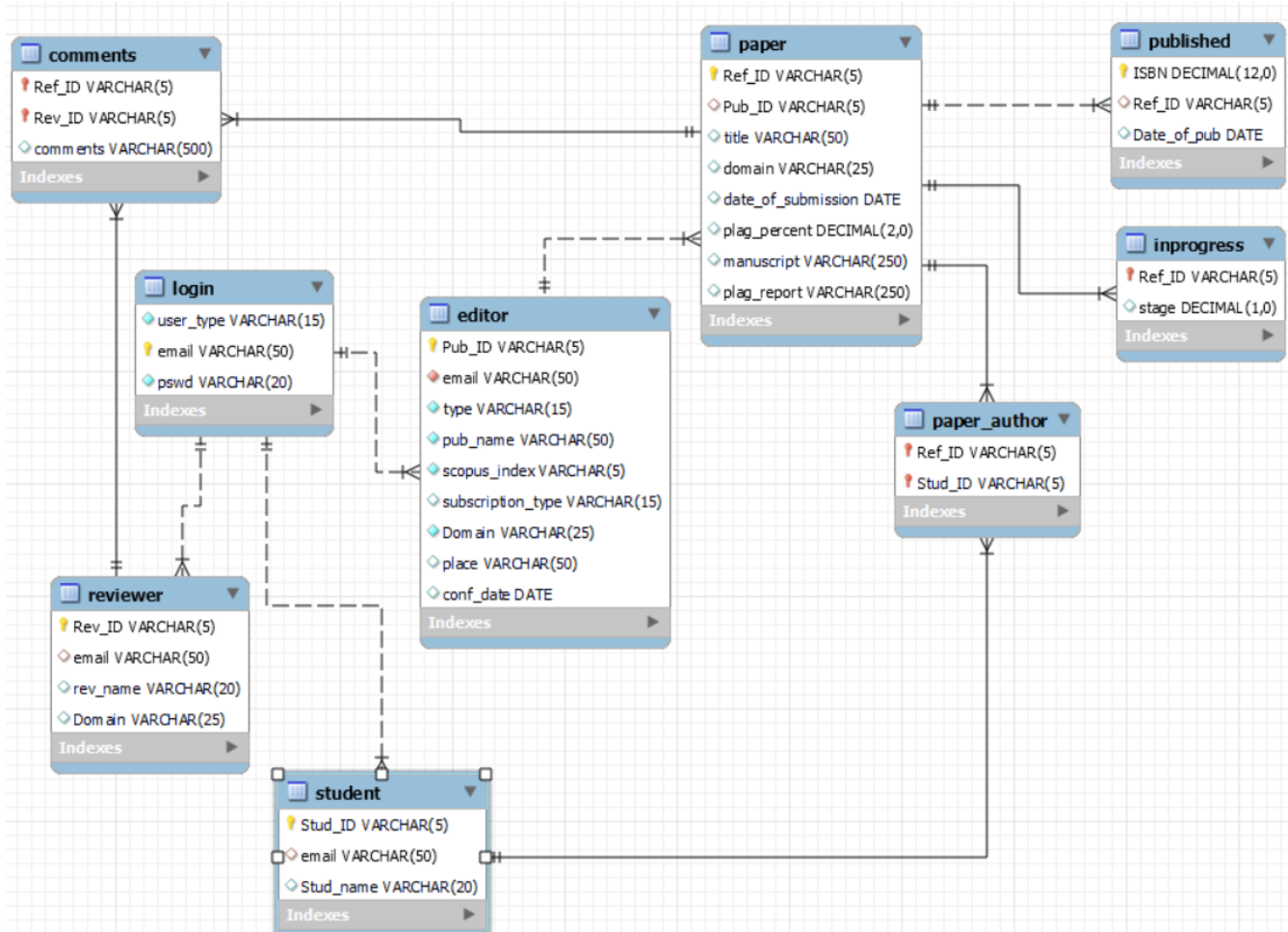
### **4.1.Hardware:**

- 4.1.1.Windows/Ubuntu/Linux/Mac OS with minimum 2GB RAM
- 4.1.2.Chrome/Firefox/Any other browser supporting HTML 5, CSS, JS
- 4.1.3.Internet Connection Required

### **4.2.Software:**

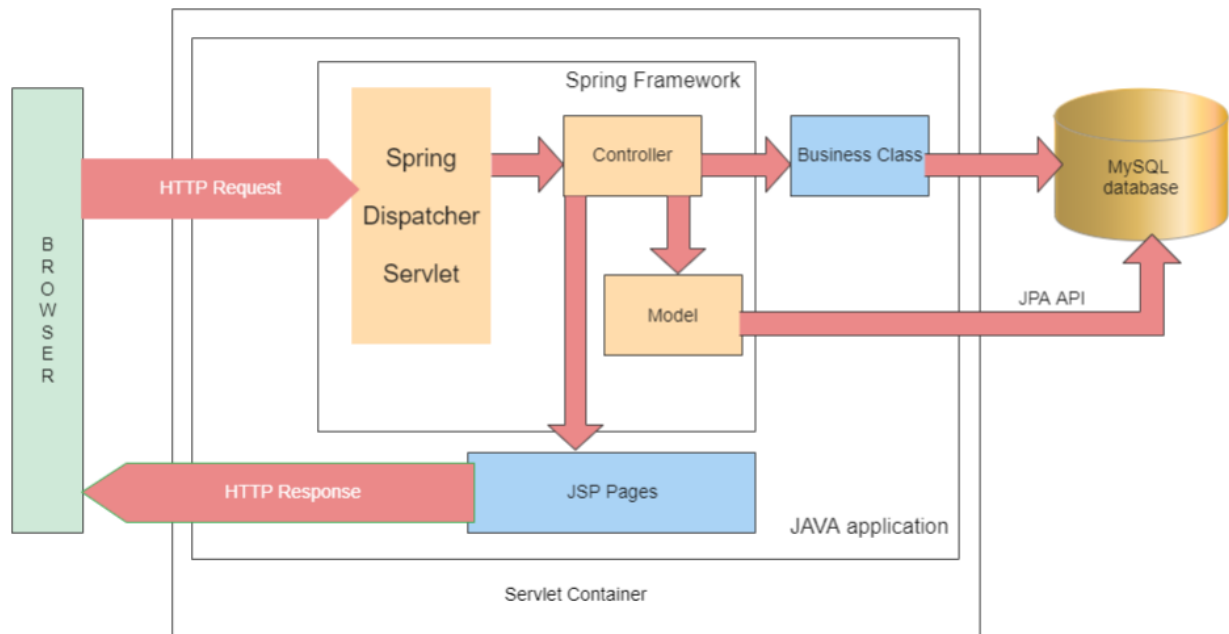
- 4.2.1. For development:
  - Visual Studio 2020
  - Browser: Any browser with HTML5 support. Chrome 92.0 or above and Opera 63 or above is recommended as they have maximum support for HTML5 tags
- 4.2.2. FRONT END- HTML, CSS
- 4.2.3. BACK END- MySQL
- 4.2.4. For Deployment
  - Operating System: Windows 7 or later, Ubuntu 20.0 or similar Linux distributions.
  - Browser: Any browser with HTML5 support. Chrome 29.0 or above and Opera 20.1 or above is recommended as they have maximum support for HTML5 tags

## 5. Schema Diagram





## 6. Architecture



## 7. Database Description

### 7.1. Paper

Table Description:

Fields:

- Ref\_ID(String)
- Pub\_ID(String)
- Title(String)
- Domain(String)
- Date\_of\_submission(Date)
- Plag\_percent(Float)
- Manuscript(String)
- Plag\_report(String)

### 7.2. Published

Table Description:

Fields:

- ISBN(Integer)
- Ref\_ID(String)
- Date\_of\_pub(Date)

### 7.3.Inprogress

Table Description:

Fields:

- Ref\_ID(String)
- Stage(Decimal)

### 7.4.Paper\_author

Table Description:

Fields:

- Ref\_ID(String)
- Stud\_ID(String)

### 7.5.Editor

Table Description:

Fields:

- Pub\_ID(String)
- Email(String)
- Type(String)
- Pub\_name(String)
- Scopus\_index(String)
- Subscription\_type(String)
- Domain(String)
- Place(String)
- Conf\_date(Date)

### 7.6.Student

Table Description:

Fields:

- Stud\_ID(String)
- Email(String)
- Stud\_name(String)

### 7.7.Login

Table Description:

Fields:

- User\_type(String)
- Email(String)
- Pswd(String)

## 7.8.Reviewer

Table Description:

Fields:

- Rev\_ID(String)
- Email(String)
- Rev\_name(String)
- Domain(String)

## 7.9.Comments

Table Description:

Fields:

- Ref\_ID(String)
- Rev\_ID(String)
- Comments(String)

8. Frontend Screen Snips
9. Servelets and Database Connectivity