

# Design Reinforcement of Beam (DRB)

Submitted for partial fulfilment of the Degree  
of  
Bachelor of Technology  
(Computer Science and Engineering)



**Submitted By:**  
Tamandeep Singh  
1410939

**Submitted To:**  
Kapil Sharma  
Training Coordinator  
CSE Department

---

Department of Computer Science & Engineering  
Guru Nanak Dev Engineering College  
Ludhiana 141006

## Acknowledgement

I, student of Guru Nanak Dev Engineering College, Ludhiana, have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

The author is highly grateful to Dr. M.S. Saini Director, Guru Nanak Dev Engineering College, Ludhiana for providing him with the opportunity to carry out his Six Weeks Training at Testing and Consultancy Cell, Guru Nanak Dev Engineering College, Ludhiana.

The author would like to whole heartedly thank Dr. H.S. Rai Dean, Testing and Consultancy Cell, Guru Nanak Dev Engineering College, Ludhiana who is a vast sea of knowledge and without whose constant and never ending support and motivation, it would never have been possible to complete the project and other assignments so efficiently and effectively.

Finally, I would thanks to all whoever have contributed in this report work with Rahul Kumar (D4 CE) and all other trainees. Without their encouragement, it would not have been possible to complete this project in such an efficient manner.

Tamandeep Singh

<b>1</b>	<b>INTRODUCTION OF ORGANIZATION</b>	<b>1</b>
1.1	Testing and Consutancy Cell . . . . .	2
<b>2</b>	<b>Introduction To Project</b>	<b>4</b>
2.1	Overview . . . . .	4
2.2	The Existing System . . . . .	4
2.3	User Requirement Analysis . . . . .	5
2.4	Feasibility Analysis . . . . .	5
2.5	Objective of Project . . . . .	6
<b>3</b>	<b>PROJECT DESIGN</b>	<b>7</b>
3.1	Software Requirement Analysis . . . . .	7
3.1.1	Functional Requirements . . . . .	7
3.1.2	Non functional requirements . . . . .	8
3.2	Dependencies . . . . .	9
<b>4</b>	<b>DEVELOPMENT AND IMPLEMENTATION</b>	<b>10</b>
4.1	Angular.js . . . . .	10
4.1.1	Features of Angular . . . . .	10
4.2	Front End Languages and Framework . . . . .	11
4.2.1	HTML . . . . .	11
4.2.2	CSS . . . . .	12
4.2.3	Javascript . . . . .	13
4.2.4	BootStarp . . . . .	13
4.3	Shell Scripting . . . . .	14
4.4	Introduction to L <sup>A</sup> T <sub>E</sub> X . . . . .	14
4.4.1	Typesetting . . . . .	15
4.5	Introduction to Node.js . . . . .	16
4.5.1	Features of Node.js . . . . .	16
4.5.2	Installation of Node.js . . . . .	16
4.5.3	Creating Node.js project . . . . .	17
4.5.4	Starting Server in node.js . . . . .	17
4.5.5	Database setup . . . . .	17

4.6	Introduction to MongoDB . . . . .	18
4.6.1	Advantages of MongnoDB . . . . .	19
4.6.2	Why to use MongnoDB . . . . .	19
4.6.3	Where to use MongnoDB . . . . .	19
4.6.4	Installation of MongoDB . . . . .	20
4.7	Introduction to Github . . . . .	20
4.7.1	What is Git? . . . . .	21
4.7.2	Installation of Git . . . . .	22
4.7.3	Various Git Commands . . . . .	22
4.7.3.1	Create Repositories . . . . .	22
4.7.3.2	Make Changes . . . . .	22
4.7.3.3	Group Changes . . . . .	23
4.7.3.4	Save Fragments . . . . .	23
4.7.3.5	Synchronize Changes . . . . .	24
4.8	Implementation . . . . .	24
<b>5</b>	<b>CONCLUSION AND FUTURE SCOPE</b>	<b>30</b>
5.1	Conclusion . . . . .	30
5.2	Future Scope . . . . .	31

## LIST OF FIGURES

1.1	Guru Nanak Dev Engineering College . . . . .	1
1.2	Testing and Consultancy Cell . . . . .	2
4.1	Angular.js logo . . . . .	10
4.2	Model View Controller . . . . .	11
4.3	HTML5 logo . . . . .	11
4.4	CSS logo . . . . .	12
4.5	Javascript logo . . . . .	13
4.6	BootStrap logo . . . . .	13
4.7	Donald Knuth, Inventor Of T <sub>E</sub> X typesetting system . . . . .	15
4.8	Node.js logo . . . . .	16
4.9	creating nodejs project.js . . . . .	17
4.10	creating server.js . . . . .	17
4.11	MongoDB logo . . . . .	18
4.12	Mean Model . . . . .	19
4.13	Github Logo . . . . .	20
4.14	Git Logo . . . . .	21
4.15	Home page of DRB . . . . .	25
4.16	Beam Views . . . . .	25
4.17	filling values . . . . .	26
4.18	resulting view . . . . .	26
4.19	resulting view . . . . .	27
4.20	resulting view . . . . .	27
4.21	Result in pdf format and other information . . . . .	28
4.22	Result in pdf format and other information . . . . .	28
4.23	Result in pdf format and other information . . . . .	29
4.24	Result in pdf format and other information . . . . .	29

	LIST OF TABLES
--	----------------

## CHAPTER 1

## INTRODUCTION OF ORGANIZATION



Figure 1.1: Guru Nanak Dev Engineering College

I had my Six Weeks Industrial Training at TCC-Testing And Consultancy Cell, GNDEC Ludhiana. Guru Nanak Dev Engineering College was established by the Nankana Sahib Education Trust Ludhiana. The Nankana Sahib Education Trust i.e NSET was founded in memory of the most sacred temple of Sri Nankana Sahib, birth place of Sri Guru Nanak Dev Ji. With the mission of Removal of Economic Backwardness through Technology Shiromani Gurudwara Parbandhak Committee i.e SGPC started a Poly technical was started in 1953 and Guru Nanak Dev Engineering College was established in 1956.

NSET resolved to uplift Rural areas by admitting 70% of students from these rural areas every year. This commitment was made to nation on 8th April, 1956, the day foundation stone of the college building was laid by Dr. Rajendra Prasad Ji, the First President of India. The College is now ISO 9001:2000 certified.

The main goal of this institute is:

- To build and promote teams of experts in the upcoming specialisations.
- To promote quality research and undertake research projects keeping in view their relevance to needs and requirements of technology in local industry.
- To achieve total financial independence.
- To start online transfer of knowledge in appropriate technology by means of establishing multipurpose resource centres.

## 1.1 Testing and Consutancy Cell

My Six Weeks Institutional Training was done by me at TCC i.e Testing And Consultancy Cell, GNDEC Ludhiana under the guidance of Dr. H.S.Rai Dean Testing and Consultancy Cell. Testing and Consultancy Cell was established in the year 1979 with a basic aim to produce quality service for technical problems at reasonable and affordable rates as a service to society in general and Engineering fraternity in particular.



Figure 1.2: Testing and Consultancy Cell

Consultancy Services are being rendered by various Departments of the College to the industry, Sate Government Departments and Entrepreneurs and are extended in the form of expert advice in design, testing of materials & equipment, technical surveys, technical audit, calibration of instruments, preparation of technical feasibility reports etc. This consultancy cell of the college has given a new dimension to the development programmers of the College. Consultancy projects of over Rs. one crore are completed by the Consultancy cell during financial year 2009-10.

Ours is a pioneer institute providing Consultancy Services in the States of Punjab, Haryana, Himachal, J&K and Rajasthan. Various Major Clients of the Consultancy Cell are as under:



- Northern Railway, Govt. of India
- Indian Oil Corporation Ltd.
- Larson & Turbo.
- Multi National Companies like AFCON & PAULINGS.
- Punjab Water Supply & Sewage Board

## CHAPTER 2

## INTRODUCTION TO PROJECT

### 2.1 Overview

Design Reinforcement of Structures (DRB) is an open-source, free web-based software, developed by students of Testing and Consultancy Cell (TCC), under the guidance of Dr. H.S. Rai. This software is used to give the Different views of structure of a building like Beams and Columns used to make a structure. This software can be used by Civil Engineers to view and analyze the structure.

The main task of this application is to get data as input from user and then it can compute the result and draw 2D views of the structure and user can download the views as a pdf format. This software is structured by keeping in view that user of this software can be both a Civil Engineer or a simple man whose job is just to enter data to software in order that an engineer can analyze later from the result stability of the structure. This software also provides the intermediate values for engineering student to deeply analyze the process of computation to be done in order to get output values.

The core part of DRB is implemented using JavaScript for processing, canvas for graphics and jspdf.js for output file (PDF) generation, Angular.js for web interface. To provide the User Experience to the users, CSS and Bootstrap has been used.

My training being not based on particular language or technology, different type of open-source softwares and technologies are used in this project and many during my training which are not used in this project like Angular.js (for a declarative user interface and as front-end), Node.js (for server-side and managing databases).

### 2.2 The Existing System

There are few existing systems for solving this particular problem like AutoCad but they don't have following features required by our mentor. These system were not open source and free web based software that were need.

All existing system suffers from at least one of the following system.

#### **Limitations of previous system**

- No batch mode

- Don't give output as PDF
- They are costly ( AutoCad costs nearly 14,000 rupees )
- They don't allow to download as pdf
- They need installation and a lot of system resources

## 2.3 User Requirement Analysis

For User Requirement Analysis, users of this system have been asked about possible requirements that this software should have and we got following resultant list of outputs-:

1. Generates the final output in the form of pdf
2. Provide on-line way to analysis so that individual does not have to install anything.
3. Save PDF in user PC in s/he favorite location
4. Make it work like batch mode. so, that user can give inputs together and relax.
5. Help B.Tech, M.Tech and Civil Engineer to analysis structure.
6. Automate calculation of modal force and modes.
7. Reduce the time for analysis.

## 2.4 Feasibility Analysis

Feasibility analysis aims to uncover the strengths and weaknesses of a project. In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility analysis should provide a historical background of the project, description of the project or service, details of the operations and management and legal requirements. Generally, feasibility analysis precedes technical development and project implementation. There is some feasibility factors by which we can determine that project is feasible or not:

- **Technical feasibility:** Technological feasibility is carried out to determine whether the project has the capability, in terms of software, hardware, personnel to handle and fulfill the user requirements. This whole project is based on solving Mathematics equations for which we have used javaScript and to provide output we have used Canvas for providing the output and jsPDF.js for pdf generation and Angular.js for user interface and Node.js as backend. Technical feasibility of this project revolves around the technical boundaries and limitations jsPDF.js and Angular.js. But as node.js is secure and structured server side framework, so these languages and technologies are perfect to design the software under this project. Design Reinforcement of Beam(DRB) is technically feasible as it is built up in Open Source Environment and thus it can be run on any Open Source platform.

- **Economic feasibility:** Economic analysis is the most frequently used method to determine the cost/benefit factor for evaluating the effectiveness of a new system. In this analysis we determine whether the benefit is gain according to the cost invested to develop the project or not. If benefits outweigh costs, only then the decision is made to design and implement the system. It is important to identify cost and benefit factors, which can be categorized as follows:
  1. Development costs.
  2. Operating costs.

Design Reinforcement of Beam(DRB) Software is also Economically feasible with 0 Development and Operating Charges as it is developed in Angular.js framework, nade.js and jspdf.js which is FOSS technology and the software is operated on Open Source platform.

- **Operational feasibility:** Operational feasibility is a measure of how well a project solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. All the Operations performed in the software are very quick and satisfies all the requirements. This project is also operational feasible as it automates the work of solving the problem of analysing the structures which not only saves time but also saves money as most of the work is done by Employees and M.Tech students is done by this software.

## 2.5 Objective of Project

Design Reinforcement of Beam(DRB) is a web based software and the main objectives of this project is to -:

1. To inspire M.Tech students to automate their work and do programming
2. Perform most of difficult Calculation work.
3. Make it work like batch mode. so, that user can give inputs together and relax.
4. Help M.Tech and Civil Engineer to analysis structure.
5. Automatic calculation of modal force and modes.
6. Reduce the time for analysis.
7. Generates the final output in the form of graphics and also in pdf format.
8. Provide on-line way to analysis so that individual does not have to install anything.
9. User can download the views in pdf file anywhere.

### 3.1 Software Requirement Analysis

A Software Requirements Analysis for a software system is a complete description of the behavior of a system to be developed. It include functional Requirements and Software Requirements. In addition to these, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation.

- **Purpose:** Dynamic of structure is a web based software and the main purpose of this project is to:
  1. Perform most of difficult Calculation work.
  2. Make it work like batch mode. so, that user can give inputs together and relax.
  3. Help M.Tech and Civil Engineer to analysis structure.
  4. Automatic calculation of modal force and modes.
  5. Reduce the time for analysis.
  6. Provide on-line way to analysis so that individual does not have to install anything.
- **Users of the System**
  1. Client : Clients are the end users that benefit from this software. They just provide input and gets output in form of PDF.Client of this WEB Application can be of two types -:
    - (a) Civil Engineer -: They have knowledge of working of procedure and what input is being provided.
    - (b) Layman -: They don't know anything about what's going on, their just work is to give input to system.

#### 3.1.1 Functional Requirements

- **Specific Requirements:** This phase covers the whole requirements for the system. After understanding the system we need the input data to the system then we watch the output

and determine whether the output from the system is according to our requirements or not. So what we have to input and then what we'll get as output is given in this phase. This phase also describes the software and non-function requirements of the system.

- **Input Requirements of the System**

1. Length of the beam
2. Width of the beam
3. Breadth of the beam
4. Diameter of the Main Bar
5. Diameter of the Anchor Bar
6. Diameter of the Stirrup Bar
7. Spacing between the stirrups

- **Output Requirements of the System**

1. Calculating and viewing graphical views of the result.
2. Generating output PDF.

- **Software Requirements**

1. Programming language: JavaScript
2. software:  $\text{\LaTeX}$
3. Libraries: jspdf.js
4. Backend: Node.js
5. Framework: Angular.js, Express.js and Bootstrap
6. Web Languages: Html, JavaScript, CSS
7. Database: MongoDB
8. Text Editor: Atom
9. Operating System: Ubuntu 14.04 or up
10. Revision System: Git

### 3.1.2 Non functional requirements

1. Scalability: System should be able to handle a number of users. For e.g., handling around thousand users at the same time.
2. Usability: Simple user interfaces that a layman can understand.
3. Speed: Processing input should be done in reasonable time i.e. we can say maximum 24 hrs.

## 3.2 Dependencies

Dependencies include softwares or framework that need to be installed for proper working of this software.

1. Programming language: JavaScript
2. Software:  $\text{\LaTeX}$
3. Framework: Angular.js and Express.js
4. Operating System: Any on which above dependencies can be installed

# CHAPTER 4

---

## DEVELOPMENT AND IMPLEMENTATION

### 4.1 Angular.js



Figure 4.1: Angular.js logo

AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. Angular's data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology.

Angular is not a single piece in the overall puzzle of building the client-side of a web application. It handles all of the DOM and AJAX glue code you once wrote by hand and puts it in a well-defined structure. This makes Angular opinionated about how a CRUD (Create, Read, Update, Delete) application should be built. But while it is opinionated, it also tries to make sure that its opinion is just a starting point you can easily change.

#### 4.1.1 Features of Angular

- It is a very good idea to decouple DOM manipulation from app logic. This dramatically improves the testability of the code.
- It is a really, really good idea to regard app testing as equal in importance to app writing. Testing difficulty is dramatically affected by the way the code is structured.
- It is an excellent idea to decouple the client side of an app from the server side. This allows development work to progress in parallel, and allows for reuse of both sides.
- It is very helpful indeed if the framework guides developers through the entire journey of building an app: From designing the UI, through writing the business logic, to testing.



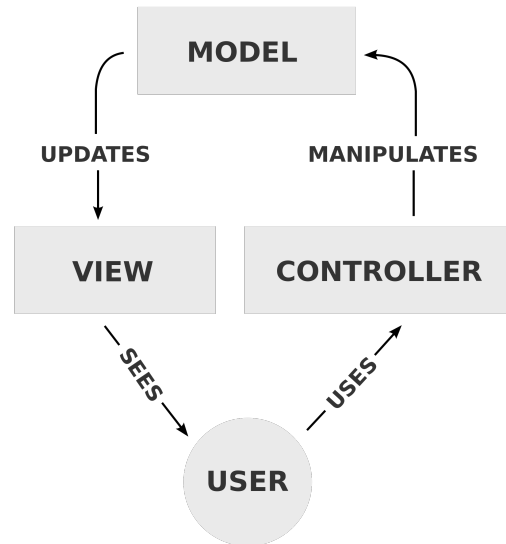


Figure 4.2: Model View Controller

- It is always good to make common tasks trivial and difficult tasks possible.

## 4.2 Front End Languages and Framework <sup>1</sup>

Front End languages are language that are used to give better user experience and user interface. These mainly include HTML, CSS, Javascript. Some Frameworks like Bootstrap are also used with these basic languages.

### 4.2.1 HTML



Figure 4.3: HTML5 logo

HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology, used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications. Web browsers can read HTML files and

<sup>1</sup> Used in project but not by me accept basic HTML

render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

### 4.2.2 CSS



Figure 4.4: CSS logo

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers. CSS makes it possible to separate presentation instructions

from the HTML content in a separate file or style section of the HTML file. For each matching HTML element, it provides a list of formatting instructions

```
p {  
    color: red;  
    text-align: center;  
}
```

### 4.2.3 Javascript



Figure 4.5: Javascript logo

JavaScript (/dvskrpt/) is a high-level, dynamic, untyped, and interpreted programming language. It has been standardized in the ECMAScript language specification. Alongside HTML and CSS, it is one of the three essential technologies of World Wide Web content production; the majority of websites employ it and it is supported by all modern web browsers without plug-ins. JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. It has an API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as networking, storage or graphics facilities, relying for these upon the host environment in which it is embedded.

### 4.2.4 BootStrap



Figure 4.6: BootStrap logo

Bootstrap is a free and open-source collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation

and other interface components, as well as optional JavaScript extensions. It aims to ease the development of dynamic websites and web applications.

Bootstrap is a front end framework, that is, an interface for the user, unlike the server-side code which resides on the "back end" or server.

### 4.3 Shell Scripting

Normally shells are interactive. It means shell accept command from you (via keyboard) and execute them. But if you use command one by one (sequence of 'n' number of commands), the you can store this sequence of command to text file and tell the shell to execute this text file instead of entering the commands. This is know as shell script. Shell script defined as series of command written in plain text file. Shell script is just like batch file is MS-DOS but have more power than the MS-DOS batch file. why to Write Shell Script ?

1. Shell script can take input from user, file and output them on screen.
2. Useful to create our own commands.
3. Save lots of time.
4. To automate some task of day-to-day life.
5. System Administration part can be also automated.

**Execute your script as syntax:**

```
chmod 755 your-script-name
sh your-script-name
./your-script-name
```

### 4.4 Introduction to L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X, I had never heard about this term before doing this project, but when I came to know about it's features, found it excellent. L<sup>A</sup>T<sub>E</sub>X (pronounced /letk/, /letx/, /ltx/, or /ltk/) is a document markup language and document preparation system for the T<sub>E</sub>X typesetting program. Within the typesetting system, its name is styled as L<sup>A</sup>T<sub>E</sub>X.

Within the typesetting system, its name is styled as L<sup>A</sup>T<sub>E</sub>X. The term L<sup>A</sup>T<sub>E</sub>X refers only to the language in which documents are written, not to the editor used to write those documents. In order to create a document in L<sup>A</sup>T<sub>E</sub>X, a .tex file must be created using some form of text editor. While most text editors can be used to create a L<sup>A</sup>T<sub>E</sub>X document, a number of editors have been created specifically for working with L<sup>A</sup>T<sub>E</sub>X.

L<sup>A</sup>T<sub>E</sub>X is most widely used by mathematicians, scientists, engineers, philosophers, linguists, economists and other scholars in academia. As a primary or intermediate format, e.g., translating DocBook and other XML-based formats to PDF, L<sup>A</sup>T<sub>E</sub>X is used because of the high quality of typesetting achievable by T<sub>E</sub>X. The typesetting system offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout and bibliographies.



Figure 4.7: Donald Knuth, Inventor Of  $\text{\TeX}$  typesetting system

$\text{\LaTeX}$  is intended to provide a high-level language that accesses the power of  $\text{\TeX}$ .  $\text{\LaTeX}$  essentially comprises a collection of  $\text{\TeX}$  macros and a program to process  $\text{\LaTeX}$  documents. Because the  $\text{\TeX}$  formatting commands are very low-level, it is usually much simpler for end-users to use  $\text{\LaTeX}$ .

#### 4.4.1 Typesetting

$\text{\LaTeX}$  is based on the idea that authors should be able to focus on the content of what they are writing without being distracted by its visual presentation. In preparing a  $\text{\LaTeX}$  document, the author specifies the logical structure using familiar concepts such as chapter, section, table, figure, etc., and lets the  $\text{\LaTeX}$  system worry about the presentation of these structures. It therefore encourages the separation of layout from content while still allowing manual typesetting adjustments where needed.

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\LaTeX}
\date{}
\begin{document}
  \maketitle
  \LaTeX{} is a document preparation system
  for the \TeX{} typesetting program.
  \par
   $E=mc^2$ 
\end{document}
```



Figure 4.8: Node.js logo

## 4.5 Introduction to Node.js

Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009 and its latest version is v0.10.36.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

### 4.5.1 Features of Node.js

- **Asynchronous and Event Driven** All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data.
- **Very Fast** Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- **Single Threaded but Highly Scalable** Node.js uses a single threaded model with event looping.
- **No Buffering** Node.js applications never buffer any data. These applications simply output the data in chunks.

### 4.5.2 Installation of Node.js

Installation of Node.js is very easy. To install Node.js, type the following commands:

```
$ curl -sL https://deb.nodesource.com/setup-6.x -o nodesource-setup.sh
```

```
$ nano nodesource-setup.sh
```

```
$ sudo apt-get install nodejs
```

```
$ sudo apt-get install build-essential
```

```
$ sudo apt-get install npm
```

This will install the node on your system.

### 4.5.3 Creating Node.js project

The source code that you would write in a source file is simply javascript. The Node.js interpreter interprets and executes your javascript code.

- **Import Required Module** We use the require directive to load the http module and store the returned HTTP instance into an http variable as follows

```
var http = require('http');

var server = http.createServer(function(request, response) {
  // magic happens here!
});
```

Figure 4.9: creating nodejs project.js

- **Create Server** We use the created http instance and call http.createServer() method to create a server instance and then we bind it at port 8081 using the listen method associated with the server instance. Pass it a function with parameters request and response. Write the sample implementation to always return "Hello World".

```
var server = http.createServer();
server.on('request', function(request, response) {
  // the same kind of magic happens here!
});
```

Figure 4.10: creating server.js

- **Testing Request & Response** Let's put step 1 and 2 together in a file called main.js and start our HTTP server. Now execute the main.js to start the server as follows

```
$ node main.js
```

### 4.5.4 Starting Server in node.js

In order to check your successful Node installation we can try out a very simple console command

```
$ node server.js
```

This command will start server in your console.

### 4.5.5 Database setup

To work with the database, you first need to create a connection. In this section of the tutorial, we will be using MongoDBs native Node.js driver to create the connection with the MongoDB server.

To install the mongodb native drivers, use the npm command to install the mongodb module. After that, run the following command in your project directory.

```
$ npm install mongodb
```

- npm is a package manager that provides a central repository for custom open source modules for Node.js and JavaScript. npm makes it simple to manage modules, their versions and distribution. As shown in the preceding paragraph, the npm install command was used to install the required module in our project.
- Load the mongodb module : We used require to load the mongodb module in our code. mongodb module represents the native mongodb drivers for Node.js.
- Defining the URL we need to connect to: We need to know where our mongodb server is running. The url represents the location where the mongodb server instance is running such that we can connect to it. The url contains the database name to which we intend to connect.
- Connect to the database: Lets use the MongoClient interface to connect to the database. In the callback we get error or the db object. We use the db object in order to communicate with the database.

## 4.6 Introduction to MongoDB



Figure 4.11: MongoDB logo

MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.



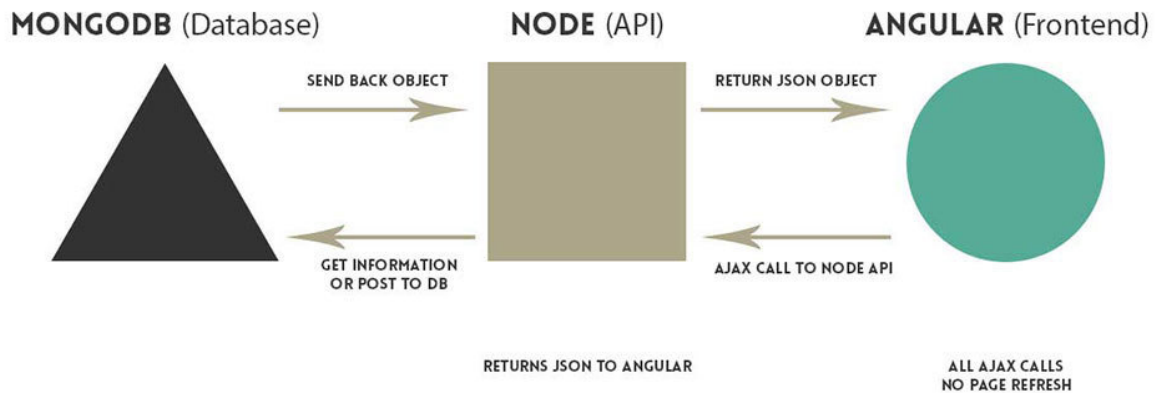


Figure 4.12: Mean Model

#### 4.6.1 Advantages of MongnoDB

- Schema less MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.
- Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
- Uses internal memory for storing the (windowed) working set, enabling faster access of data.
- Conversion/mapping of application objects to database objects not needed.

#### 4.6.2 Why to use MongnoDB

- Document Oriented Storage Data is stored in the form of JSON style documents.
- Index on any attribute.
- Replication and high availability.
- Auto-sharding

#### 4.6.3 Where to use MongnoDB

- Big Data
- Content Management and Delivery
- Mobile and Social Infrastructure
- User Data Management

#### 4.6.4 Installation of MongoDB

Doxygen can be installed using following commands:

```
$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 0C49F3730359A14518585931BC777777
$ sudo apt-get update
$ sudo apt-get install -y mongodb-org
$ sudo service mongod start
```

### 4.7 Introduction to Github



Figure 4.13: Github Logo

GitHub is a Git repository web-based hosting service which offers all of the functionality of Git as well as adding many of its own features. Unlike Git which is strictly a command-line tool, Github provides a web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project.

GitHub offers both paid plans for private repto handle everything from small to very large projects with speed and efficiency. ositories, and free accounts, which are usually used to host open source software projects. As of 2014, Github reports having over 3.4 million users, making it the largest code host in the world.

GitHub has become such a staple amongst the open-source development community that many developers have begun considering it a replacement for a conventional resume and some employers require applications to provide a link to and have an active contributing GitHub account in order to qualify for a job.

The Git feature that really makes it stand apart from nearly every other Source Code Management (SCM) out there is its branching model.

Git allows and encourages you to have multiple local branches that can be entirely independent of each other. The creation, merging, and deletion of those lines of development takes seconds.

This means that you can do things like:

- Frictionless Context Switching.  
Create a branch to try out an idea, commit a few times, switch back to where you branched from, apply a patch, switch back to where you are experimenting, and merge it in.
- Role-Based Code lines.  
Have a branch that always contains only what goes to production, another that you merge work into for testing, and several smaller ones for day to day work.
- Feature Based Work flow.  
Create new branches for each new feature you're working on so you can seamlessly switch back and forth between them, then delete each branch when that feature gets merged into your main line.
- Disposable Experimentation.  
Create a branch to experiment in, realize it's not going to work, and just delete it - abandoning the work with nobody else ever seeing it (even if you've pushed other branches in the meantime).

Notably, when you push to a remote repository, you do not have to push all of your branches. You can choose to share just one of your branches, a few of them, or all of them. This tends to free people to try new ideas without worrying about having to plan how and when they are going to merge it in or share it with others.

There are ways to accomplish some of this with other systems, but the work involved is much more difficult and error-prone. Git makes this process incredibly easy and it changes the way most developers work when they learn it.

### 4.7.1 What is Git?



Figure 4.14: Git Logo

Git is a distributed revision control and source code management (SCM) system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Git was initially designed and developed by Linus Torvalds for Linux kernel development in 2005, and has since become the most widely adopted version control system for software development.

As with most other distributed revision control systems, and unlike most clientserver systems, every Git working directory is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server. Like the Linux kernel, Git is free and open source software distributed under the terms of the GNU General Public License version 2 to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

### 4.7.2 Installation of Git

Installation of git is a very easy process. The current git version is: 2.0.4. Type the commands in the terminal:

```
$ sudo apt-get update
```

```
$ sudo apt-get install git
```

This will install the git on your pc or laptop.

### 4.7.3 Various Git Commands

Git is the open source distributed version control system that facilitates GitHub activities on your laptop or desktop. The commonly used Git command line instructions are:-

#### 4.7.3.1 Create Repositories

Start a new repository or obtain from an exiting URL

```
$ git init [ project-name ]
```

Creates a new local repository with the specified name

```
$ git clone [url ]
```

Downloads a project and its entire version history

#### 4.7.3.2 Make Changes

Review edits and craft a commit transaction

```
$ git status
```

Lists all new or modified files to be committed

```
$ git diff
```

Shows file differences not yet staged

**\$ git add [file ]**

Snapshots the file in preparation for versioning

**\$ git reset [file ]**

Unstages the file, but preserve its contents

**\$ git commit -m [descriptive message ]**

Records file snapshots permanently in version history

#### 4.7.3.3 Group Changes

Name a series of commits and combine completed efforts

**\$ git branch**

Lists all local branches in the current repository

**\$ git branch [branch-name ]**

Creates a new branch

**\$ git checkout [branch-name ]**

Switches to the specified branch and updates the working directory

**\$ git merge [branch ]**

Combines the specified branches history into the current branch

**\$ git branch -d [branch-name ]**

Deletes the specified branch

#### 4.7.3.4 Save Fragments

Shelve and restore incomplete changes

**\$ git stash**

Temporarily stores all modified tracked files

**\$ git stash pop**

Restores the most recently stashed files

**\$ git stash list**

Lists all stashed changesets

**\$ git stash drop**

Discards the most recently stashed changeset

#### 4.7.3.5 Synchronize Changes

Register a repository bookmark and exchange version history

**\$ git fetch [bookmark ]**

Downloads all history from the repository bookmark

**\$ git merge [bookmark /[branch]]**

Combines bookmarks branch into current local branch

**\$ git push [alias [branch]]**

Uploads all local branch commits to GitHub

**\$ git pull**

Downloads bookmark history and incorporates changes

## 4.8 Implementation

Development of DRB started with development in phases which focus on particular need of project. Various phases and their detail are given below -:

- Phase I :  
During Phase I, we wrote code in JavaScript to compute all the required output variable. We use different controllers for the application. Controller named BeamControl fetch the data from the user and perform operations and gives out required result.
- Phase II (L<sup>A</sup>T<sub>E</sub>X) :  
During Phase II, we wrote code in JavaScript to compute all the required output variable. We use different controllers for the application. Controller named BeamControl fetch the data from the user and perform operations and gives out required result.
- Phase III :  
During phase III, we provided web interface to this software using Django. Djanog was used to get input from user and write input.sage file for particular user then civil.sh is called by passing name of user directory to it and then get output PDF.
- Phase IV :  
During phase IV, we improved the code structure and added additional functionality like sending PDF as email and accepting input as CSV file. Finally, the UI was improved and made responsive.
- Phase V :  
During phase V, we tested the software for various conditions and then applied required error control and messaging mechanism. initialfile.py file was created to save software from problem of server restart which can causes processing user request to stop. so, that the interrupted request of user can be restart and send PDF.
- Phase VI:-  
During final phase, we documented the project( developers documentation and README.md) using doxygen and wrote the report for this software.

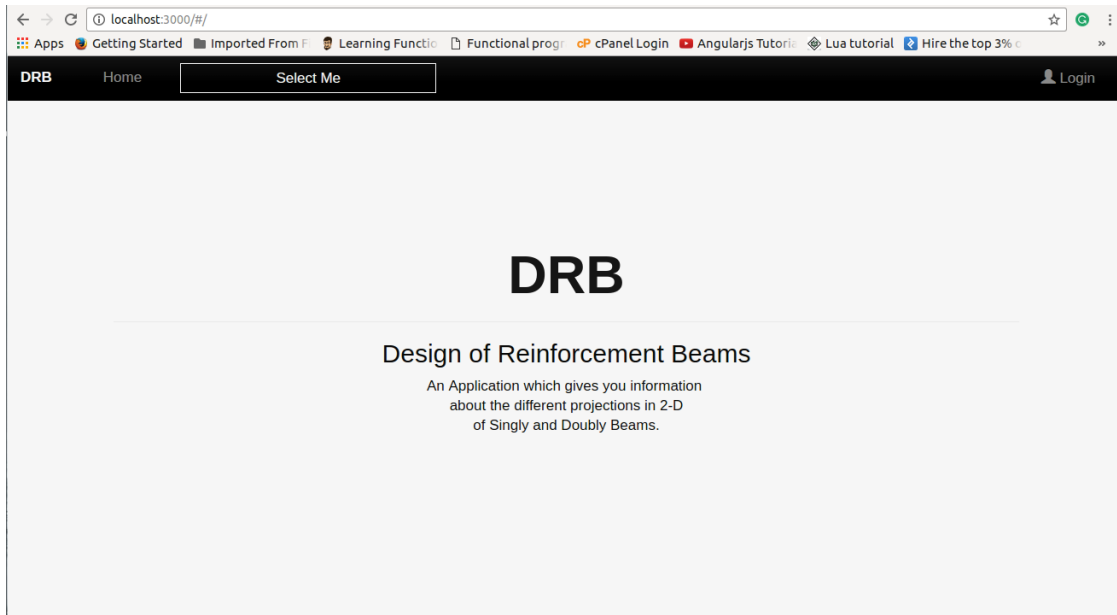


Figure 4.15: Home page of DRB

This is the first interface that is shown to the user. This is the Home page. It brings a nice UI to get input from user about the Number of storeys and some factor affecting the structure. It also provides the user some additional features as an aid like moving to specific operations the user want to do. There is an icon of logo i.e DRB which directs the user to home. There is also an login icon for user to login from different clients like gmail and facebook.4.15

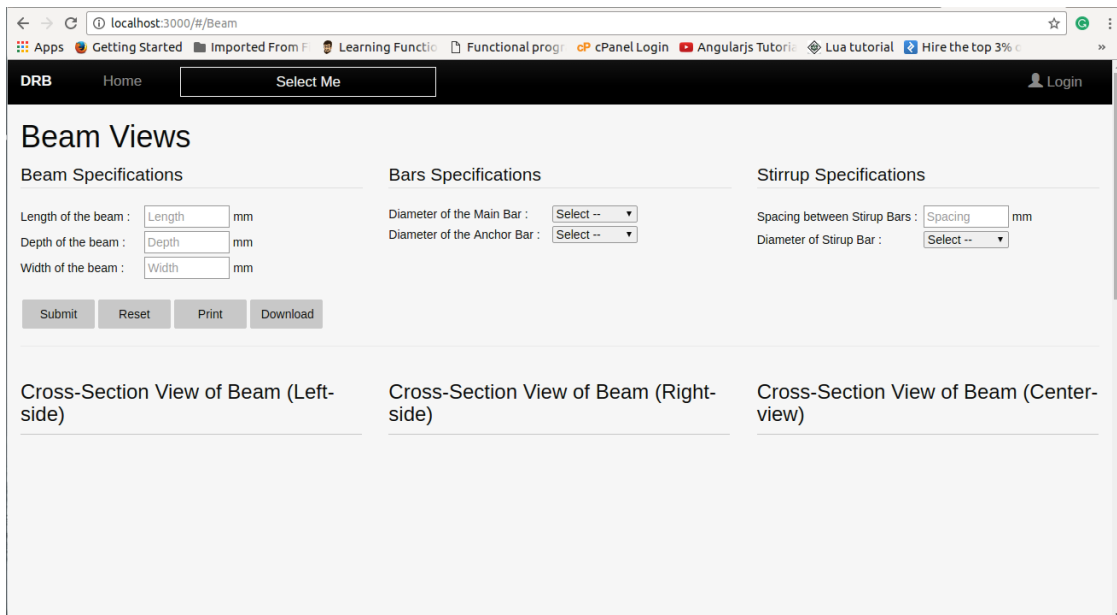


Figure 4.16: Beam Views

After filling the required data click the 'Submit' button on the page, Different views are drawn under the specific places.4.16

The screenshot shows a web browser at localhost:3000/#/Beam. The application has a dark header with 'DRB', 'Home', and a 'Select Me' button. A 'Login' link is in the top right. The main content area is titled 'Beam Views' and contains three columns of input fields:

- Beam Specifications:**
  - Length of the beam : 3000 mm
  - Depth of the beam : 600 mm
  - Width of the beam : 300 mm
- Bars Specifications:**
  - Diameter of the Main Bar : 25 mm
  - Diameter of the Anchor Bar : 12 mm
- Stirrup Specifications:**
  - Spacing between Stirup Bars : 100 mm
  - Diameter of Stirup Bar : 8 mm

Below the input fields are four buttons: 'Submit', 'Reset', 'Print', and 'Download'. At the bottom, there are three empty boxes labeled 'Cross-Section View of Beam (Left-side)', 'Cross-Section View of Beam (Right-side)', and 'Cross-Section View of Beam (Center-view)'.

Figure 4.17: filling values

After filling certain values.4.17

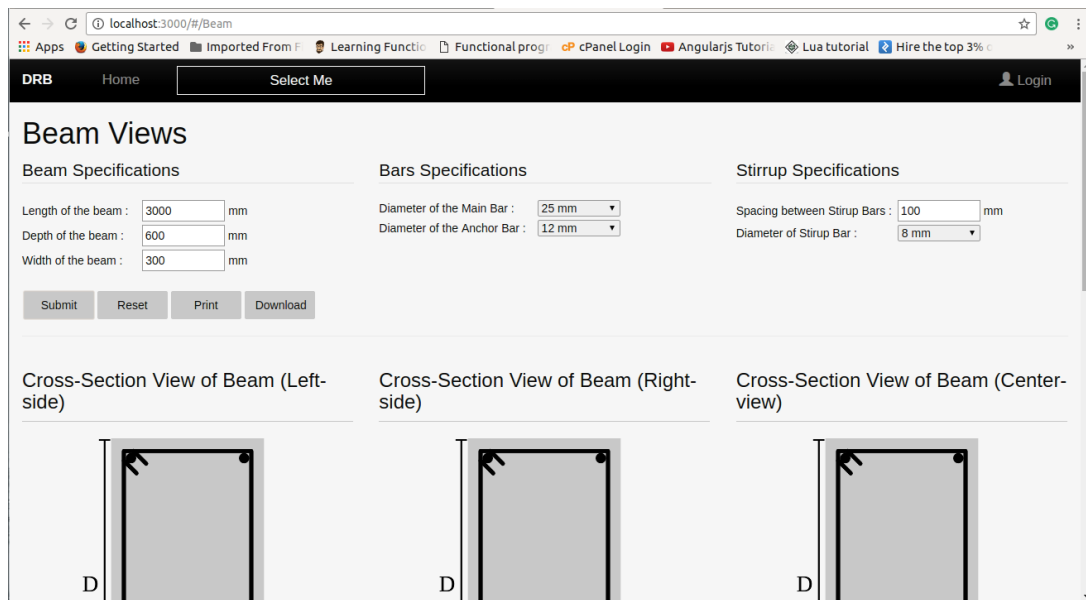


Figure 4.18: resulting view



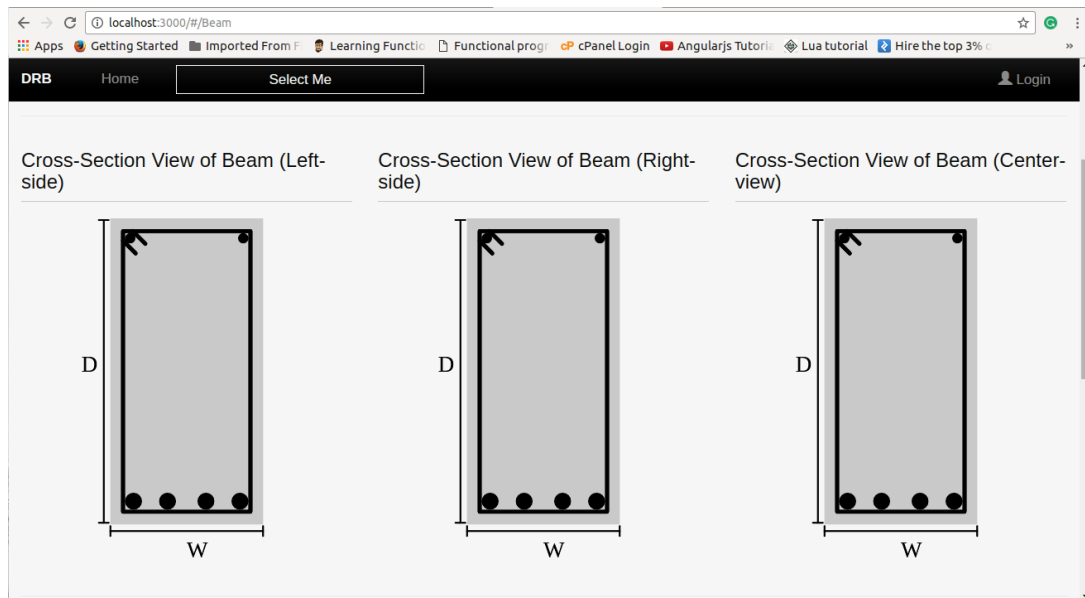


Figure 4.19: resulting view

Output of the result Cross section view of beam(left, right and center)4.184.19

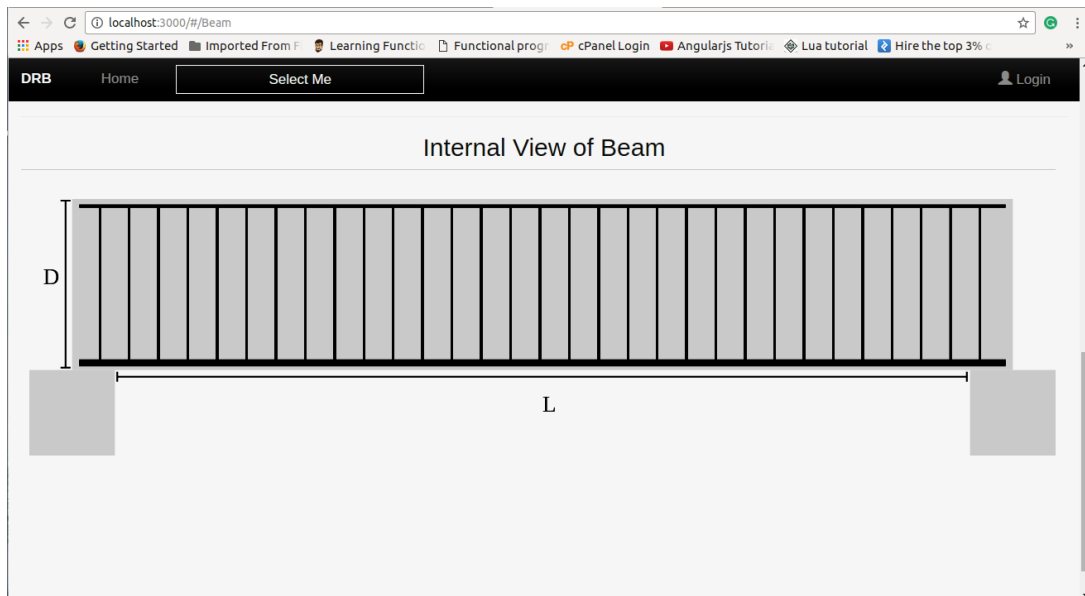


Figure 4.20: resulting view

Output of the Internal view of the beam.4.20

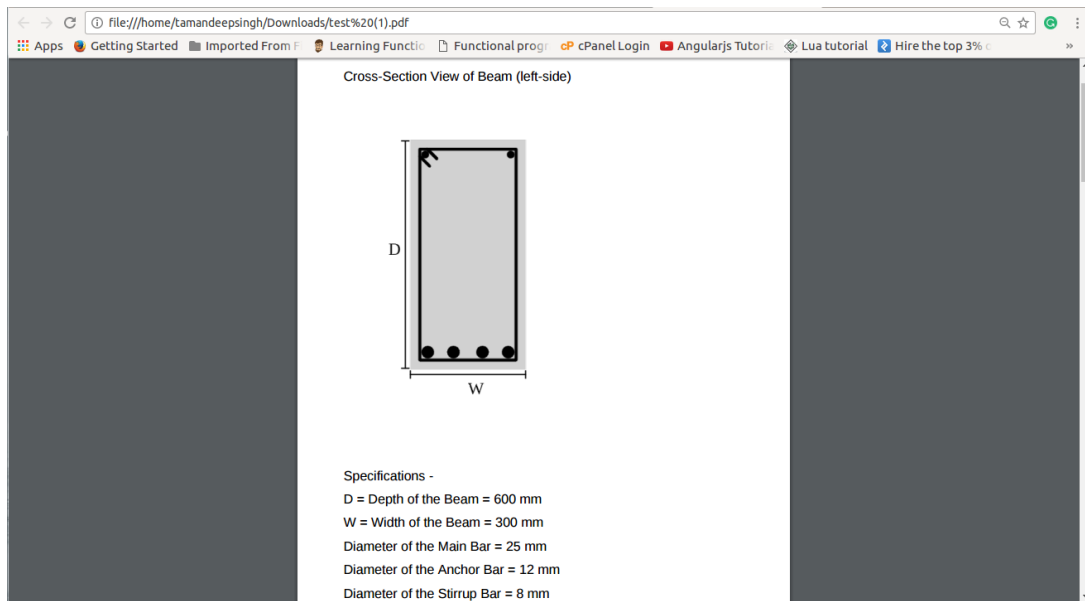


Figure 4.21: Result in pdf format and other information

Output View of cross section of beam (Left side)4.21.

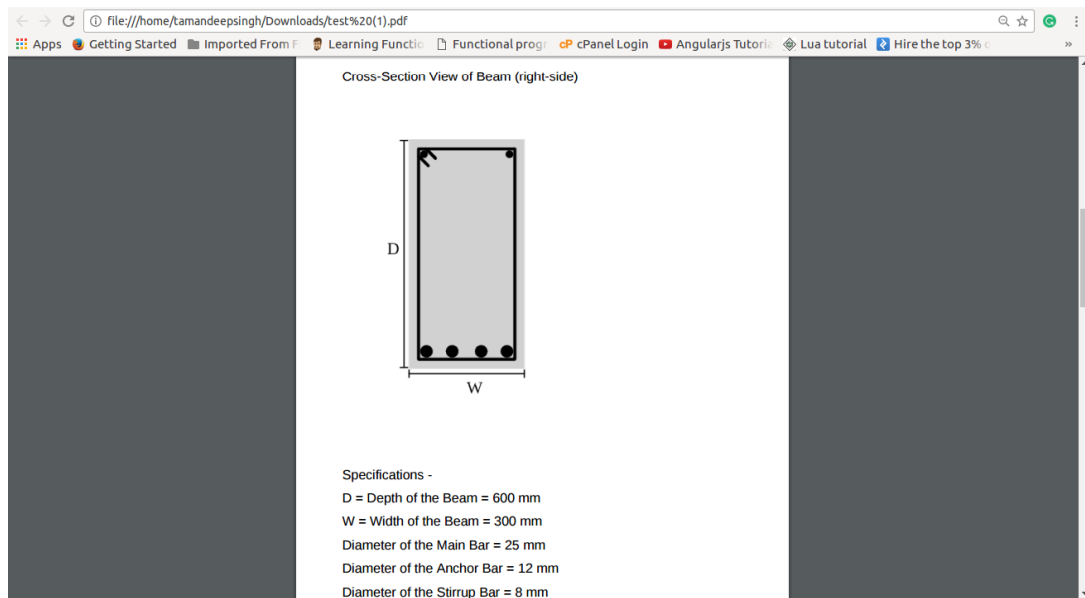


Figure 4.22: Result in pdf format and other information

Output View of cross section of beam (Right side) 4.22.

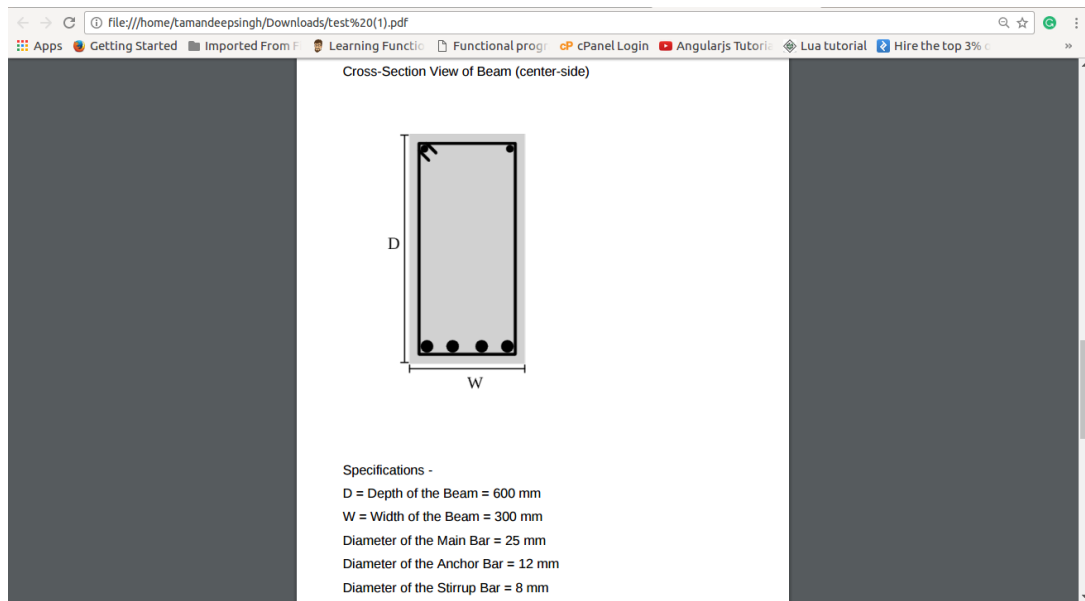


Figure 4.23: Result in pdf format and other information

Output View of cross section of beam (Center side)4.23.

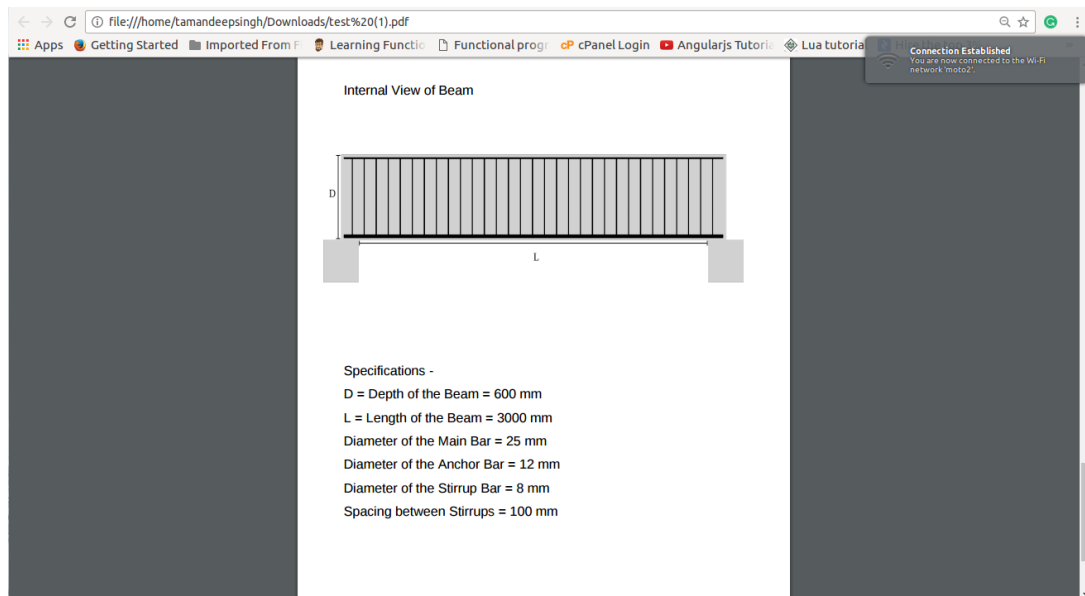


Figure 4.24: Result in pdf format and other information

Output View of Internal view of the Beam4.24.

## 5.1 Conclusion

DRB is a very efficient application which help in generating result for analysis of structures. It can be used by Civil Engineers and M.Tech. and B.Tech students and even layman. It's less time consuming and user-friendly and free from all the installation process. It is a web application that can be accessed from a number of devices. The responsive User Interface makes it easy for the users to operate it. Many efforts were made to ease the usage for the users. Hence, it is expected to be work properly in different conditions. But any future bug reports or improvements are always welcomed and will be processed happily.

I learn a lot by working on this project. During this period I got to learn a vast number of technologies. These are listed below: Operating system: Ubuntu Language used: HTML, CSS and javaScript Framework: Angular.js and Express.js Technogloy: jspdf.js, Doxygen, Git So during this project I learn all the above things. Above all I got to know how software is developed and how much work and attention to details is required in building even the most basic of components of any project. Planning, designing, developing code, working in a team, testing, etc. These are all very precious lessons in themselves. Aside from all above I got go know about various methods like -:

1. Threading the programs
2. Embedding and using different tech in one software.
3. How to work like in group for development of software.
4. How to apply juggaar(innovated) in softwares to get problem solved.

Beside these technology used in project I also get to know some other tech also like -:

1. opensshserver
2. reveal.js, impress.js (for making presentations)

## 5.2 Future Scope

This software being a open source have a lot of scope for future improvements and additions as other individuals can also contribute in it and add additional functionality like-:

1. Output real time graph of modes
2. Give user other analyses option
3. Add ability to model structure online
4. Automating other analyses techniques
5. Taking minimum amount of data and giving required result

- [1] Design Reinforcement of Beam, <https://github.com/TamandeepSingh/DRB>
- [2] L<sup>A</sup>T<sub>E</sub>X <https://www.sharelatex.com>
- [3] Node.js <https://nodejs.org/en/docs/>
- [4] Angular.js <https://docs.angularjs.org/guide/introduction>
- [5] My Blog, <http://tamandeepsingh.wordpress.com/>
- [6] My Github Profile, <https://github.com/TamandeepSingh>