

**B.Tech Project Report**

**ADMIN PORTAL**  
**BACHELOR OF TECHNOLOGY IN**  
**INFORMATION TECHNOLOGY**

**Submitted By:**

Nitish Kumar (2K19/IT/093)

Naveen Kumar(2K19/IT/087)

**Under the Guidance Of:**

Ms Swati Sharda



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**DELHI TECHNOLOGICAL UNIVERSITY**

**(Formerly Delhi College of Engineering), Bawana Road,**

**Delhi-110042**

**MAY 2021**

**CANDIDATE'S DECLARATION**

We, hereby declare that the project work entitled “Admin Portal” submitted by **Nitish Kumar(2K19/IT/093)** and **Naveen Kumar (2K19/IT/087)** to Department Of Information Technology, Delhi Technological University is a record of bonfide Project work carried out by us under the guidance of **Ms Swati Sharda** and this project is submitted in the partial fulfilment of the requirements for the awards of the degree of Bachelor of Technology in Information Technology. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any diploma or degree.

Delhi

Naveen Kumar

Date: May 14,2021

Nitish Kumar

**CERTIFICATE**

This is to certify that the project entitled, “**Admin Portal**” submitted by **Nitish Kumar(2K19/IT/093)** and **Naveen Kumar (2K19/IT/087)** in partial fulfilment of the requirements for the award of Bachelor of Technology in Information Technology under the guidance of **Ms. Swati Sharda** is an authentic work carried out by them under my supervision and guidance. To the best of my knowledge all the work has been done on their own and has not been copied from elsewhere without proper citation.

Delhi

Ms. Swati Sharda

Date: May,15,2021

## **ACKNOWLEDGEMENT**

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to our professor Ms. Swati Sharda for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude towards our parents & member of Delhi Technological University for their kind co-operation and encouragement which help us in completion of this project.

We would like to express our special gratitude and thanks to industry persons for giving us such attention and time.

Our thanks and appreciations also go to my colleague in developing the project and people who have willingly helped us out with their abilities.

## **Abstract:**

The project is an admin portal of a website for a society/club which organizes workshop, competitions and events for students/professionals. It is a data management system which can be used for storing data of team members of the society/club and data of various events, workshops and competitions organized by the club. This website will allow users to perform all the basic operations required to manage Data in the database like:

- 1) adding data to the database
- 2) removing data from the database
- 3) updating data in the database.
- 4) retrieving data from the database based upon a priority.
- 5) searching data in the database.

# **Project related Links:**

## **1. GitHub repository**

<https://github.com/Nitish9711/AdminPanel>

## **2. Website**

<https://dbms-project-admin-portal.herokuapp.com/>

## **3. Windows Setup**

[https://drive.google.com/file/d/1iNJeoW\\_Gzqcv5yx8HFjk3kuq04pEcl8/view?usp=sharing](https://drive.google.com/file/d/1iNJeoW_Gzqcv5yx8HFjk3kuq04pEcl8/view?usp=sharing)

## **4. Windows Zip folder**

<https://drive.google.com/file/d/1x9LI589TEnHGdghd3RXGRVj7DB9a3kHC/view?usp=sharing>

## **5. Linux Zip folder**

<https://drive.google.com/file/d/1-AhtostuyN6mMZr5OO-ZwIEyyJG6A7Yq/view?usp=sharing>

## **TABLE OF CONTENTS**

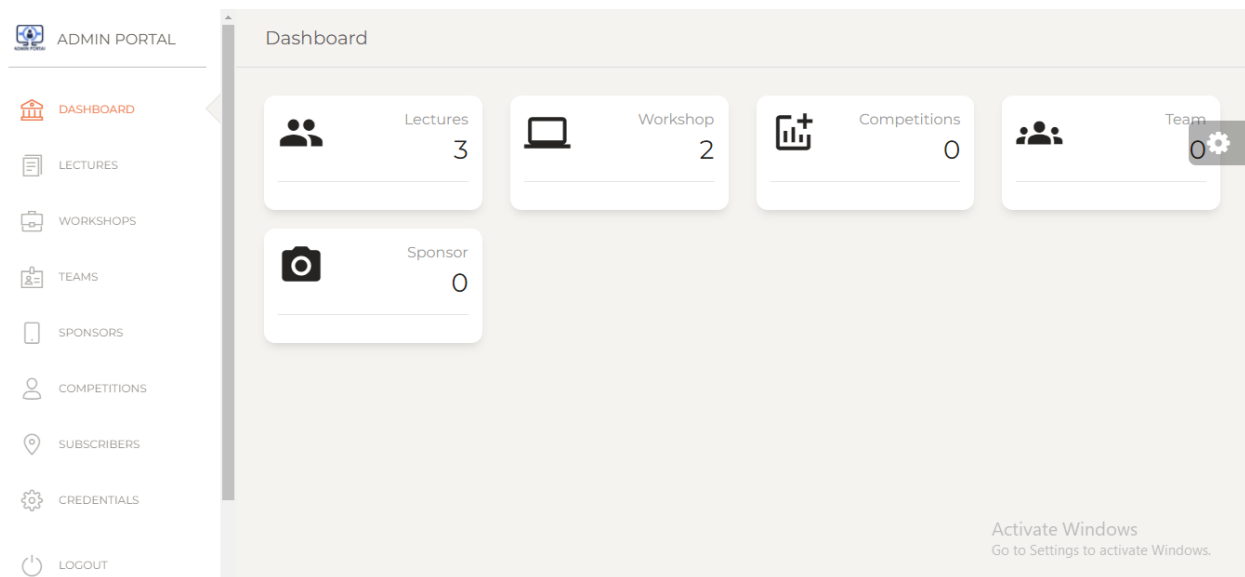
Abstract	5
Project related Links	6
Table Of Contents	7
<b>Chapter 1: Introduction</b> <ul style="list-style-type: none"> <li>• 1.1 Introduction To admin Portal <ul style="list-style-type: none"> <li>➤ 1.1.1 Purpose of the Project</li> <li>➤ 1.1.3 Scope of the Project.</li> <li>➤ 1.1.4 Bottlenecks of Existing System.</li> </ul> </li> </ul>	8 8 9 9
<b>Chapter 2: Programming Tools</b> <ul style="list-style-type: none"> <li>• 2.1 MONGODB <ul style="list-style-type: none"> <li>➤ 2.1.1 What is MONGODB?</li> <li>➤ 2.1.2 Rich JSON Documents</li> <li>➤ 2.1.3 Powerful query language</li> </ul> </li> <li>• 2.2 EXPRESS.JS <ul style="list-style-type: none"> <li>➤ 2.2.1 Features of EXPRESS.JS</li> </ul> </li> <li>• 2.3 ANGULAR <ul style="list-style-type: none"> <li>➤ 2.3.1 A complete client side solution</li> <li>➤ 2.3.2 The Zen of AngularJS</li> </ul> </li> <li>• 2.4 NODE.JS <ul style="list-style-type: none"> <li>➤ 2.4.1 What is node.js?</li> <li>➤ 2.4.2 What can node.js do?</li> <li>➤ 2.4.2 What is a node.js file?</li> </ul> </li> <li>• 2.5 ELECTRON.JS</li> <li>• 2.6 HEROKU</li> </ul>	10 10 11 11 11 12 13-14 14 14 14 14 15 15
<b>Chapter 3: Graphical User Interface</b> <ul style="list-style-type: none"> <li>• 3.1 Login Window</li> <li>• 3.2 dashboard</li> <li>• 3.3 Lectures and other sections</li> </ul>	16 17 17
<b>Chapter 4: Conclusion</b>	18
<b>Chapter 5: References</b>	19

# Chapter 1: Introduction

---

## 1.1 Introduction to Admin Portal

It is a data management system which can be used for storing data of team members of the society/club and data of various events, workshops and competitions organized by the club. This website will allow users to perform all the basic operations required to manage Data in the database.



### 1.1.1 PURPOSE OF PROJECT:

The purpose of this project is to make management of data easy for a society/club that organizes a lot of events, workshops and competitions for students and professionals. It can be used by the club to manage the details of their events, workshops, competitions, lectures and sponsors in a fast and efficient manner. We have created a user-friendly interface using



which one can add new data to the database, remove data from the database and fetch data from the database

### **1.1.3 SCOPE OF THE PROJECT:**

This project will be actually used for managing the website of a society. The admin portal that we have made will be used as an administration console from which one can manage various identity services.

### **1.1.4 Bottlenecks of the Existing System:**

The existing system has lot of problems such as:

- ✓ The entire database is maintained manually which is rather tedious and error prone.
- ✓ More space is required to keep all the records.
- ✓ Improper interface.
- ✓ Inefficiency in the whole system.

# Chapter 2: Programming Tools

---

## MEAN STACK :

**MEAN** (MongoDB, Express.js, AngularJS (or Angular), and Node.js) is a free and open-source JavaScript software stack for building dynamic web sites and web applications.<sup>[2]</sup>

Because all components of the MEAN stack support programs that are written in JavaScript, MEAN applications can be written in one language for both server-side and client-side execution environments.

## 2.1 MONGODB

### 2.1.1 What is MONGODB?

MongoDB is a document database, which means it stores data in JSON-like documents. We believe this is the most natural way to think about data, and is much more expressive and powerful than the traditional row/column model.

### 2.1.2 Rich JSON Documents

- The most natural and productive way to work with data.
- Supports arrays and nested objects as values.
- Allows for flexible and dynamic schemas.

### 2.1.3 Powerful query language

- Rich and expressive query language that allows you to filter and sort by any field, no matter how nested it may be within a document.
- Support for aggregations and other modern use-cases such as geo-based search, graph search, and text search.
- Queries are themselves JSON, and thus easily composable. No more concatenating strings to dynamically generate SQL queries.

## 2.2 EXPRESS JS

Express.js (also referred to as Express) is a modular web application framework package for Node.js.

While Express is capable of acting as an internet-facing web server, even supporting SSL/TLS out of the box, it is often used in conjunction with a reverse proxy such as NGINX or Apache for performance reasons

### 2.2.1 Features Of EXPRESS.JS:

1. Faster Server side development
2. Middleware
3. Routing
4. Templating
5. Debugging

## 2.3 ANGULAR

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. AngularJS'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.

AngularJS is what HTML would have been, had it been designed for applications. HTML is a great declarative language for static documents. It does not contain much in the way of creating applications, and as a result building web application is an exercise in *what do I have to do to trick the browser into doing what I want?*

The impedance mismatch between dynamic applications and static documents is often solved with:

- **a library** - a collection of functions which are useful when writing web apps. Your code is in charge and it calls into the library when it sees fit. E.g., jQuery.
- **frameworks** - a particular implementation of a web application, where your code fills in the details. The framework is in charge and it calls into your code when it needs something app specific. E.g., durandal, ember, etc.

AngularJS takes another approach. It attempts to minimize the impedance mismatch between document centric HTML and what an application needs by creating new HTML constructs. AngularJS teaches the browser new syntax through a construct we call *directives*. Examples include:

- Data binding, as in `{{}}`.
- DOM control structures for repeating, showing and hiding DOM fragments.
- Support for forms and form validation.
- Attaching new behavior to DOM elements, such as DOM event handling.
- Grouping of HTML into reusable components.

### 2.3.1 A complete client-side solution

AngularJS 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 AngularJS 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.

AngularJS comes with the following out-of-the-box:

- Everything you need to build a CRUD app in a cohesive set: Data-binding, basic templating directives, form validation, routing, deep-linking, reusable components and dependency injection.
- Testability story: Unit-testing, end-to-end testing, mocks and test harnesses.
- Seed application with directory layout and test scripts as a starting point.

### 2.3.2 The Zen of AngularJS

AngularJS is built around the belief that declarative code is better than imperative when it comes to building UIs and wiring software components together, while imperative code is excellent for expressing business logic.

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

AngularJS frees you from the following pains:

- **Registering call-backs:** Registering call-backs clutters your code, making it hard to see the forest for the trees. Removing common boilerplate code such as call-backs is a good thing. It vastly reduces the amount of JavaScript coding *you* have to do, and it makes it easier to see what your application does.
- **Manipulating HTML DOM programmatically:** Manipulating HTML DOM is a cornerstone of AJAX applications, but it's cumbersome and error-prone. By declaratively describing how the UI should change as your application state changes, you are freed from low-level DOM manipulation tasks. Most applications written with AngularJS never have to programmatically manipulate the DOM, although you can if you want to.
- **Marshaling data to and from the UI:** CRUD operations make up the majority of AJAX applications' tasks. The flow of marshaling data from the server to an internal object to an HTML form, allowing users to modify the form, validating the form, displaying validation errors, returning to an internal model, and then back to the server, creates a lot of boilerplate code. AngularJS eliminates almost all of this boilerplate, leaving code that describes the overall flow of the application rather than all of the implementation details.
- **Writing tons of initialization code just to get started:** Typically, you need to write a lot of plumbing just to get a basic "Hello World" AJAX app working. With AngularJS you can bootstrap your app easily using services, which are auto-injected into your application in a Guice-like dependency-injection style. This allows you to get started developing features quickly. As a bonus, you get full control over the initialization process in automated tests.

## 2.4 NODE.JS

### 2.4.1 What is Node.js?

- Node.js is an open-source server environment
- Node.js is free
- Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- Node.js uses JavaScript on the server

## 2.4.2 What Can Node.js Do?

- Node.js can generate dynamic page content
- Node.js can create, open, read, write, delete, and close files on the server
- Node.js can collect form data
- Node.js can add, delete, modify data in your database

## 2.4.3 What is a Node.js File?

- Node.js files contain tasks that will be executed on certain events
- A typical event is someone trying to access a port on the server
- Node.js files must be initiated on the server before having any effect
- Node.js files have extension ".js"

## 2.5 ELECTRON.JS

**Electron** (formerly known as **Atom Shell**) is an open-source software framework developed and maintained by GitHub. It allows for the development of desktop GUI applications using web technologies: it combines the Chromium rendering engine and the Node.js runtime. Electron is the main GUI framework behind several open-source projects including Atom, GitHub Desktop, Light Table, Visual Studio Code, Evernote, and WordPress Desktop .

## 2.6 HEROKU

**Heroku** is a cloud platform as a service (PaaS) supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages. Heroku was acquired by Salesforce.com in 2010 for \$212 million.

# Chapter 3: Graphical User Interface

---

## 3.1 LOGIN:

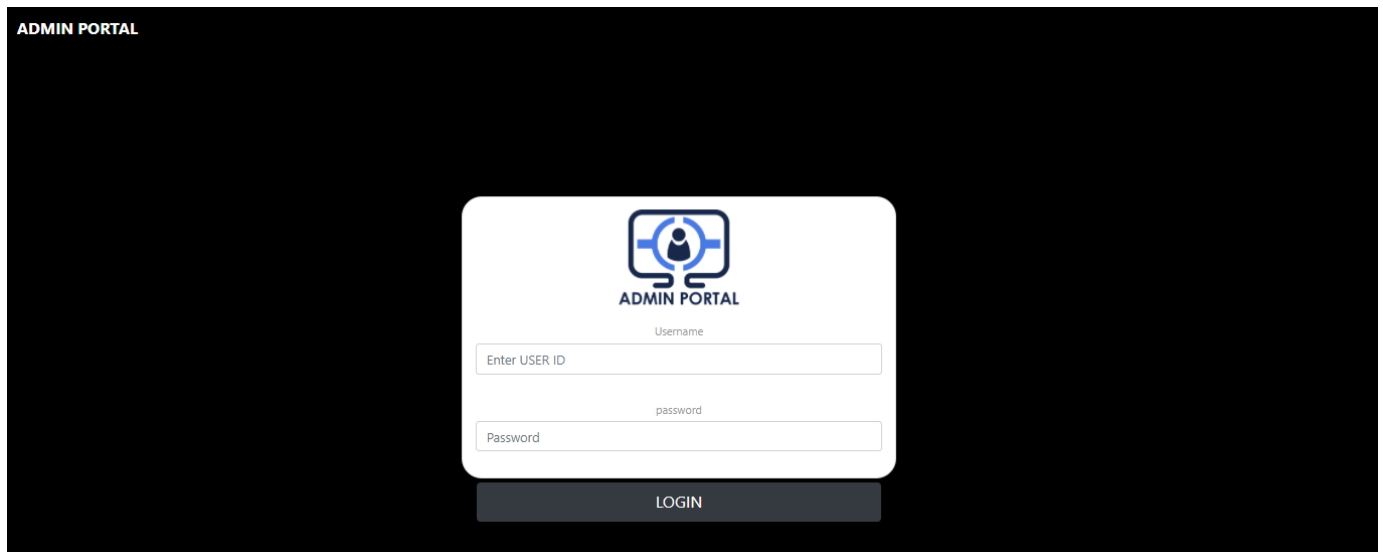
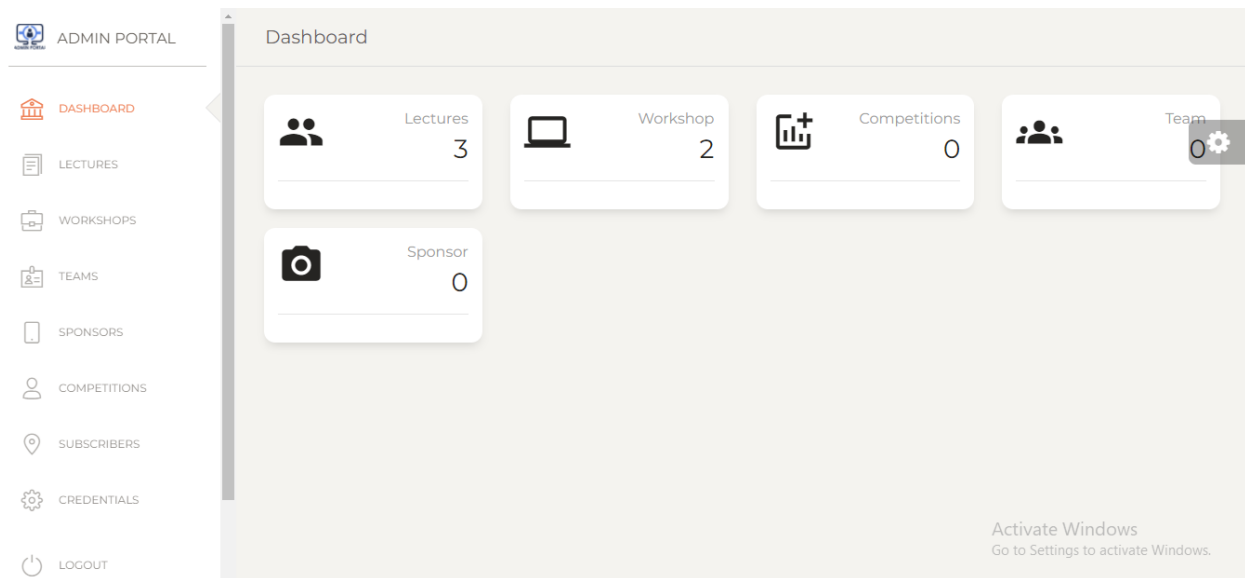


Fig. 3.1: Login Window



## 3.2 DashBoard:



## 3.3 LECTURES:

The screenshot shows the 'Lectures' section of the Admin Portal. It features a 'Lecture List' table with a header row and three data rows. Each row includes a thumbnail image preview and action buttons for edit and delete. A sidebar on the left contains an 'Add Lecture' button and a settings gear icon. At the bottom right, there is a message: 'Activate Windows Go to Settings to activate Windows.'

Name	Profession	Lecture Title	Date	time	Status	Registration Link	Image Preview	Action
demo	demo	demo	1/4/2021	demo	upcoming	demo		
daf	adsfad	dasfad	1/4/2021	adfas	upcoming	adfds		
dafds	dafa	daf	2/4/2021	adf	upcoming	dafs		

# Chapter 4

---

## **CONCLUSION**

The entire project has been developed and deployed as per the requirements stated by the user, it is found to be bug free as per the testing standards that is implemented. Any specification-untraced errors will be concentrated in the coming versions, which are planned to be developed in near future. The system at present does not take care off the money payment methods, as the consolidated constructs need SSL standards and are critically to be initiated in the first face, the application of the credit card transactions is applied as a developmental phase in the coming days. The system needs more elaborative technicality for its inception and evolution.

# Chapter 5

---

## REFERENCES

- <https://www.electron.build/>
- <https://docs.angularjs.org/guide>
- <https://www.udemy.com/course/angular-2-and-nodejs-the-practical-guide/>
- <https://www.electronjs.org/docs>
- <https://mongoosejs.com/docs/documents.html>

**THANK YOU!**