**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

"Jnana Sangama", Belagavi: 590018



Internship report on

**“WEB DEVELOPER”**

A Dissertation work submitted in partial fulfillment of the requirement for the award of the degree of

**Bachelor of Engineering**

**in**

**Computer Science and Engineering**

by

**R Gagan Bharadwaj 1AY15CS073**

Under the guidance of

**Dr. P V Kumar**

Professor

|  |  |
| --- | --- |
| |  | | --- | |  | |



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**ACHARYA INSTITUTE OF TECHNOLOGY**

(Affiliated to Visvesvaraya Technological University, Belagavi)

**2018-2019**

**ACHARYA INSTITUTE OF TECHNOLOGY**

Acharya Dr. Sarvepalli Radhakrishnan Road, Soladevanahalli, Bangalore – 560107

(Affiliated to Visvesvaraya Technological University, Belagavi)

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**Certificate**

Certified that the internship entitled **“Web Developer”** is a bonafide work carried out by **R Gagan Bharadwaj (1AY15CS073)** in partial fulfillment for the award of degree of **Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University**, Belgaum during the year **2018-2019.** It is certified that all corrections/ suggestions indicated for internal assessments have been incorporated in the Report deposited in the departmental library. The Internship report has been approved as it satisfies the academic requirements in respect of Internship prescribed for the **Bachelor of Engineering Degree**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signature of Guide**  Dr. P V Kumar  Professor |  | **Signature of H.O.D**  Dr. Prashanth C M  Head of the Department |  | **Signature of Principal**  Dr.  Prakash M R  Principal |

**Signature of Coordinator**

Dr. V Nagaveni

Professor

|  |  |  |  |
| --- | --- | --- | --- |
| **External Viva** | | | |
|  | **Name of the Examiners** |  | **Signature with Date** |
|  |  |  |  |
|  |  |  |  |

**ACKNOWLEDGEMENT**

I express my gratitude to my institution and management for providing me with good infrastructure, laboratory, facilities and inspiring staff, and whose gratitude was of immense help in completion of this report successfully.

I deeply indebted to **Dr.  Prakash M R,** Principal, Acharya Institute of Technology, Bangalore, who has been a constant source of enthusiastic inspiration to steer me forward.

I hearty thank **Dr. Prashanth C M**, Head of the Department, Department of Computer Science and Engineering, Acharya Institute of Technology Bangalore, for his valuable support and for rendering resources for this Internship work.

I specially thank **Dr. P V Kumar** Professor, Department of Computer Science and Engineering who guided me with valuable suggestions in completing this Internship at every stage.

Also, I wish to express deep sense of gratitude for Internship coordinator Dr. **V Nagaveni** Professor, Department of Computer Science and Engineering, Acharya Institute of Technology for her support and advice during the course of this final year internship.

I would like to express my sincere thanks and heartfelt gratitude to my beloved Parents, Respected Professors, Classmates, Friends, Juniors for their indispensable help at all times.

Last but not the least my respectful thanks to the Almighty.

**R Gagan Bharadwaj (1AY15CS073)**

**ABSTRACT**

The objective of our Internship is to develop a responsive modern application for the web, mobile, and desktop by using Angular 6 framework.

Angular is one of the powerful and widely used frameworks for modern single-page web application development which is designed to support dynamic views in the applications. This framework is based on the model-view-controller pattern to construct single-page web apps, as the scripting part is done using TypeScript which is an advancement of JavaScript, and has HTML5 along with all the new CSS types (like SCSS, SASS, etc.….).

During the internship period I was part of the development team led by technology solution architect in developing a web-based learning management system. I have worked on the frontend design and implementation for both user end and admin end. I have further worked on integrating social and payment gateways to the system. I have worked on setting up the interaction between frontend modules and backend with Http requests and responses using httpClient and other npm packages.

CONTENTS

**Sl. No. Chapter Name Page No.**

1. **Chapter 1 – About the industry 1**
   1. Introduction 2
   2. About Company 2
2. **Chapter 2–Training/Technology 3**
   1. Angular 4
   2. NodeJS 4
   3. MongoDB 4
3. **Chapter 3– System Design 5**
   1. Angular 6 6
   2. NodeJS 8
   3. ExpressJS 9
   4. MongoDB 10
4. **Chapter 4 –Implementation 12**
5. **Chapter 5– Results 15**
6. **Chapter 6– Learning Outcomes 17**
7. **Details of stipend 19**

Chapter 1

about the industry

# **Chapter 1**

# **about the industry**

**1.1 Introduction:**

Initiated by alumnus of IIMs, Business Toys Pvt Ltd is an Edu-tech startup focused on enhancing online learning experience through Gamification.

Business Toys offers cloud-based learning platform to provide training and assessment services for skills like Business analytics, financial Analytics, Marketing Analytics education.

**1.2 About Company:**

"Learning Celebrated" is what we aim to achieve in every classroom across India. Business Toys Pvt. Ltd, seeks to bring in sheer "FUN" into classroom by way of web-based simulations for academic theories, concepts & models. We believe in "PLAY > LEARN > GET AHEAD" as a model. Currently the target Market is academic institutions & students of Plus two, Graduation and Post-Graduation in commerce and Management stream.”

-Business Toys

Chapter 2

Training

/

Technology

# **Chapter 2**

# **Training/technology**

**2.1 Angular:**

Angular is a platform that makes it easy to build applications with the web. Angular combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges. Angular empowers developers to build applications that live on the web, mobile, or the desktop.

**2.2 NodeJS:**

"Node.js is a JavaScript runtime environment”. JavaScript now has the capability to do things that other scripting languages like Python can do with the help of NodeJS.

Node.js came into existence when the original developers of JavaScript extended it from something you could only run in the browser to something you could run on your machine as a standalone application. Both your browser JavaScript and Node.js run on the V8 JavaScript runtime engine. This engine takes your JavaScript code and converts it into a faster machine code. Machine code is low-level code which the computer can run without needing to first interpret it.

**2.3 MongoDB:**

“MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need”.

MongoDB is an open source database management system (DBMS) that uses a document-oriented database model which supports various forms of data. It is one of numerous nonrelational database technologies which arose in the mid-2000s under the NoSQL banner for use in big data applications and other processing jobs involving data that doesn't fit well in a rigid relational model. Instead of using tables and rows as in relational databases, the MongoDB architecture is made up of collections and documents.

Chapter 3

software Design

# **Chapter 3**

# **Software Design**

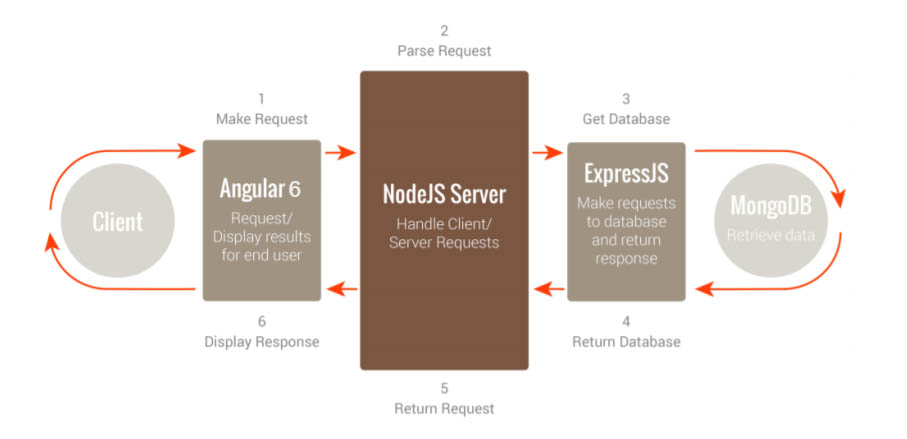
****

Fig 3.1: Working of MEAN stack software’s/technologies.

Above fig 3.1 shows the flow of data throughout the software.

* When client makes any request, first it will be processed by the Angular 6. Angular 6 is a client-side language in JavaScript.
* After that the Request enters in phase 2 which is NodeJS. NodeJS is a server-side language in JavaScript.
* After that Request enter in the phase 3 which is ExpressJS it makes request to the database.
* After that MongoDB retrieve the data and return the response to the ExpressJS.
* Then ExpressJs return response to the NodeJS and then NodeJS return it to the AngularJS to display the result.

**3.1 Angular 6:**

Angular 6 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 data binding and dependency injection eliminates much of the code you currently have to write. And it all happens within the browser, making it an ideal partner with any server technology.

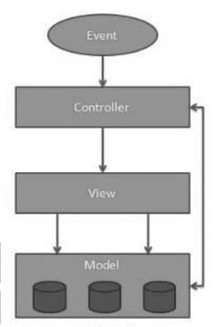
****

Fig 3.1.1: Angular 6 Archietucture.

**Model :**

It is the lowest level of the pattern responsible for maintaining data.

The model is responsible for managing application data. It responds to the request from view and to the instructions from the controller to update itself.

**View :**

It is responsible for displaying all or a portion of the data to the user.

A presentation of data in a particular format, triggered by the controller’s decision to present the data. They are script-based template systems such as JSP, ASP, PHP and very easy to integrate with AJAX technology.

**Controller :**

It is a software Code that controls the interactions between the Model and View.

The controller responds to user input and performs interactions on the data model objects. The controller receives input, validates it, and then performs business operations that modify the state of the data model.

**3.2 NodeJS:**

Node.js is a server-side platform built on Chrome’s JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

**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. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call

**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. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.

**No Buffering :**

Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.

**3.3 ExpressJS:**

Express provides a minimal interface for us to build our applications. It is minimal, providing us the absolutely required tools to build our app and flexible, there are numerous modules available on npm for express, which can be directly plugged into express.

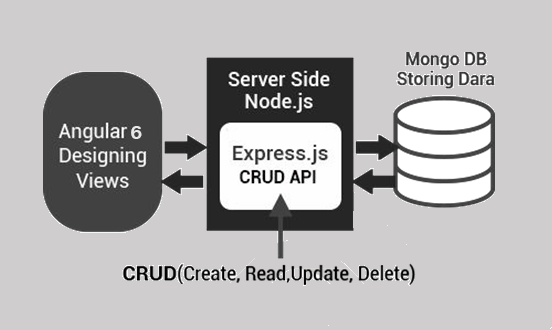
Unlike its competitors like Rails and Django, which have an opinionated way of building applications, express has no “best way” do something. It is very flexible and pluggable.

Fig 3.3.1: Architecture of ExpressJS.

When the user sends a request to through AngularJS then that request is firstly accessed by the NodeJS threading is done in the NodeJS and then it is sent to the ExpressJS to Create, Read, Update and Delete the API for the Request. ExpressJs host the website for the NodeJS. Both NodeJS and ExpressJS are server side languages. After CURD the API data is retrieved from the MongoDB ad then send it to the User.

* Create (POST) – Make something
* Read (GET)\_- Get something
* Update (PUT) – Change something
* Delete (DELETE)- Remove something

**3.4 MongoDB:**

MongoDB (from humongous) is a free and open-source, cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. MongoDB is the database for today’s applications, enabling you to:

* Leverage data and technology to maximize competitive advantage
* Reduce risk for mission-critical deployments
* Accelerate time-to-value
* Dramatically lower total cost of ownership

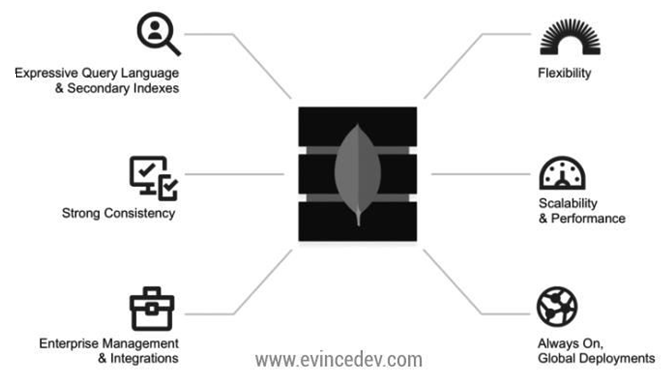


Fig 3.3.1: Architecture of MongoDB.

**Expressive query language & secondary Indexes :**

Users should be able to access and manipulate their data in sophisticated ways to support both operational and analytical applications. Indexes play a critical role in providing efficient access to data, supported natively by the database rather than maintained in application code.

**Strong consistency :**

Applications should be able to immediately read what has been written to the database. It is much more complex to build applications around an eventually consistent model, imposing significant work on the developer, even for the most sophisticated engineering teams.

**Enterprise Management and Integrations :**

Databases are just one piece of application infrastructure and need to fit seamlessly into the enterprise IT stack. Organizations need a database that can be secured, monitored, automated, and integrated with their existing technology infrastructure, processes and staff, including operations teams, DBAs, and data analysts.

**Flexible Data Model :**

NoSQL databases emerged to address the requirements for the data we see dominating modern applications. Whether document, graph, key-value, or wide-column, all of them offer a flexible data model, making it easy to store and combine data of any structure and allow dynamic modification of the schema without downtime or performance impact.

**Scalability and Performance :**

NoSQL databases were all built with a focus on scalability, so they all include some form of shading or partitioning. This allows the database to scale out on commodity hardware deployed on-premises or in the cloud, enabling almost unlimited growth with higher throughput and lower latency than relational databases.

**Always-On Global Deployments :**

* NoSQL databases are designed for highly available systems that provide a consistent, high-quality experience for users all over the world. They are designed to run across many nodes, including
* replication to automatically synchronize data across servers, racks, and data centers.

Chapter 4

implementation

# **Chapter 4**

# **implementation details**

All the collected information about the avilable courses, quizes, compitations, hackathons are arranged in the form of card as shown in below fig 4.1.

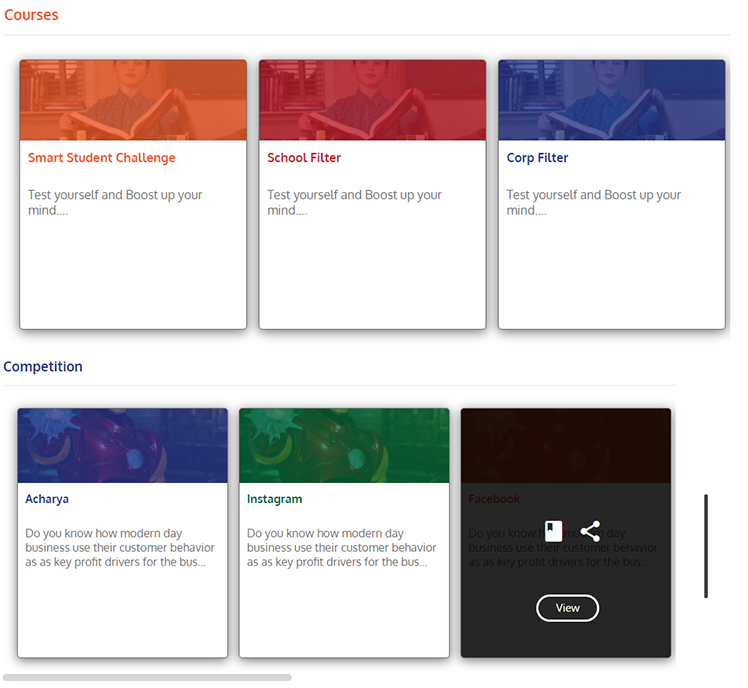


Fig 4.1: Card style arrangement for the available data.

As the user first requires to login to access the avilable informations. The login form is shown in below fig 4.2(a), and the fig 4.2(b) is the signup form.

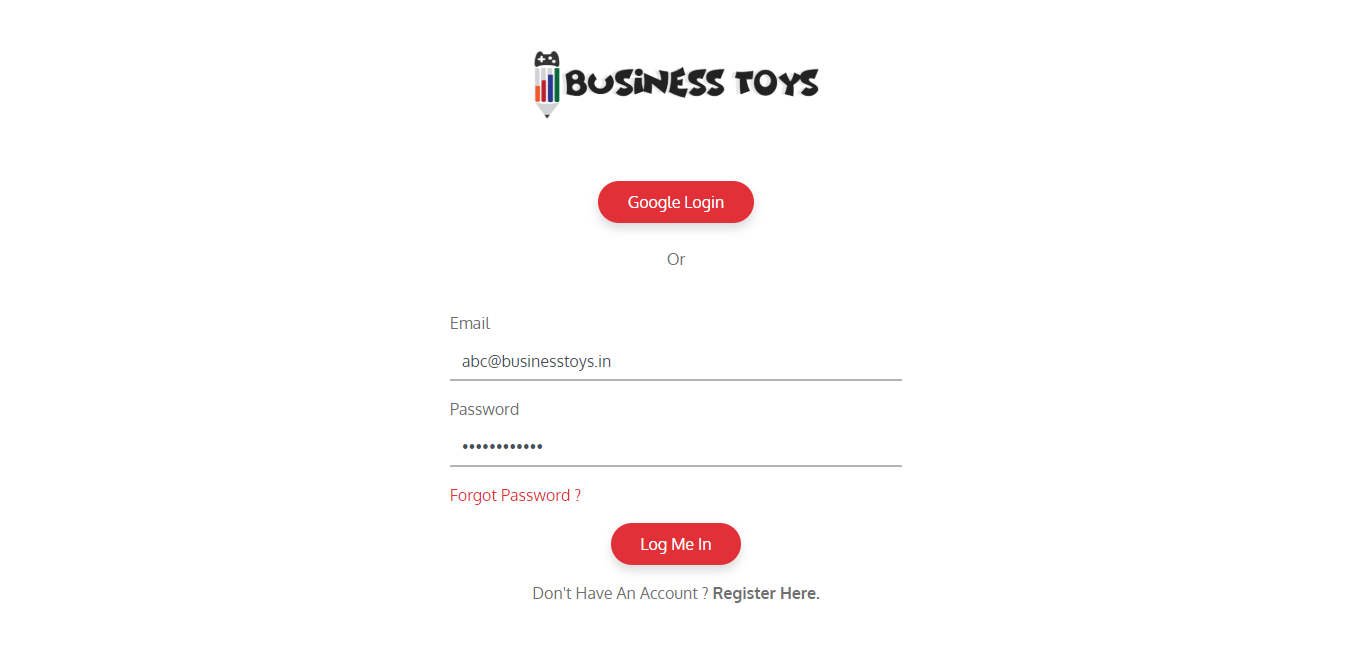


Fig 4.2(a): login form.

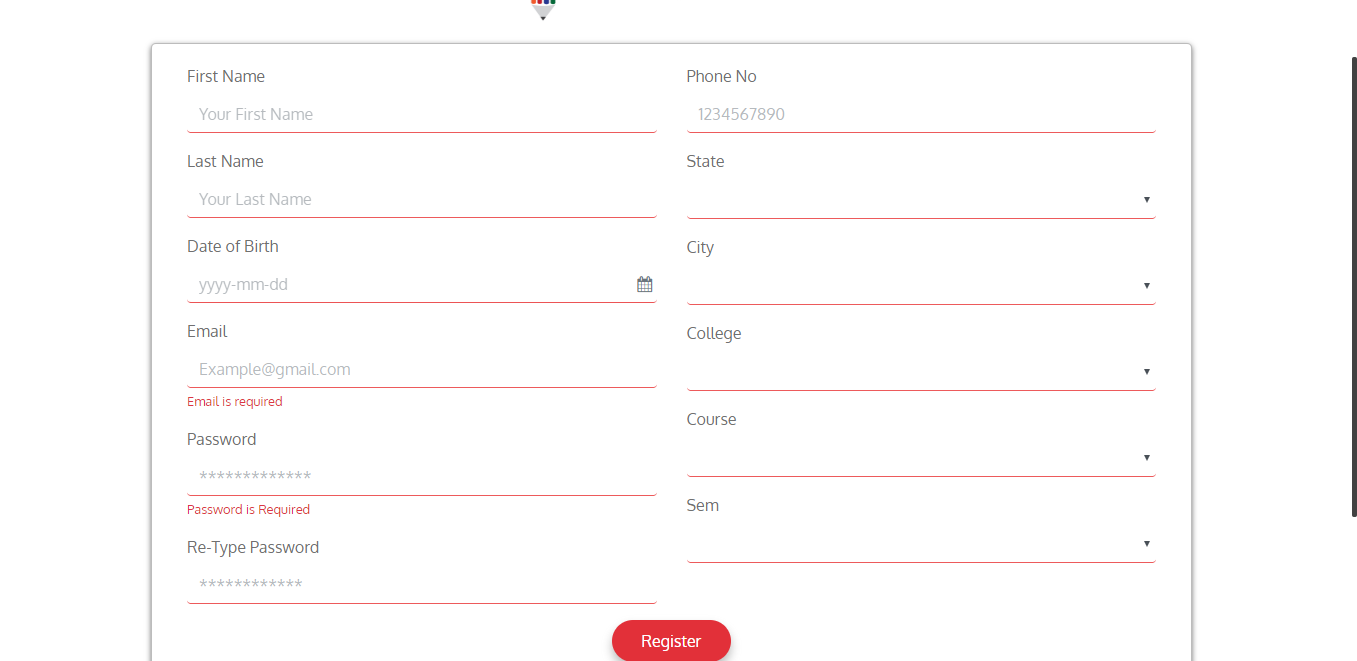


Fig 4.2(b): signup form.

When the user logs into the site, series of angular routing will procedes as the they proceeds from home page to Cirtification page and back to home page again. The series of pages follows as home – courses detail – payment gateway – Simulation of the selected course – games based on selected things – quiz – score board – cirtification.

Chapter 5

Results

# **Chapter 5**

# **Results**

I majorly worked on developing Admin panel, along with client side (front end) for the product which is shown in fig 5.1(a) and fig 5.1(b). That process works internally as we know the following client-based information will be depending on the data which admin provides. And the finished product works as we expected and we hosted through AWS server.

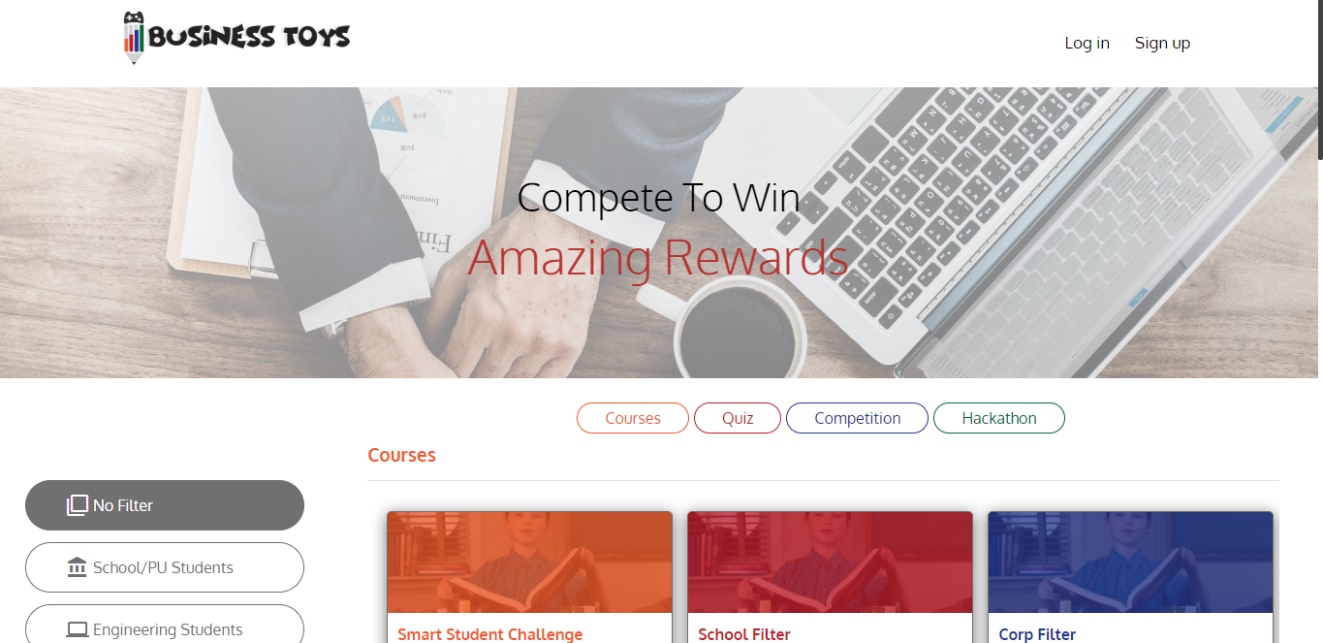
****

Fig 5.1(a): Home page of the Product.

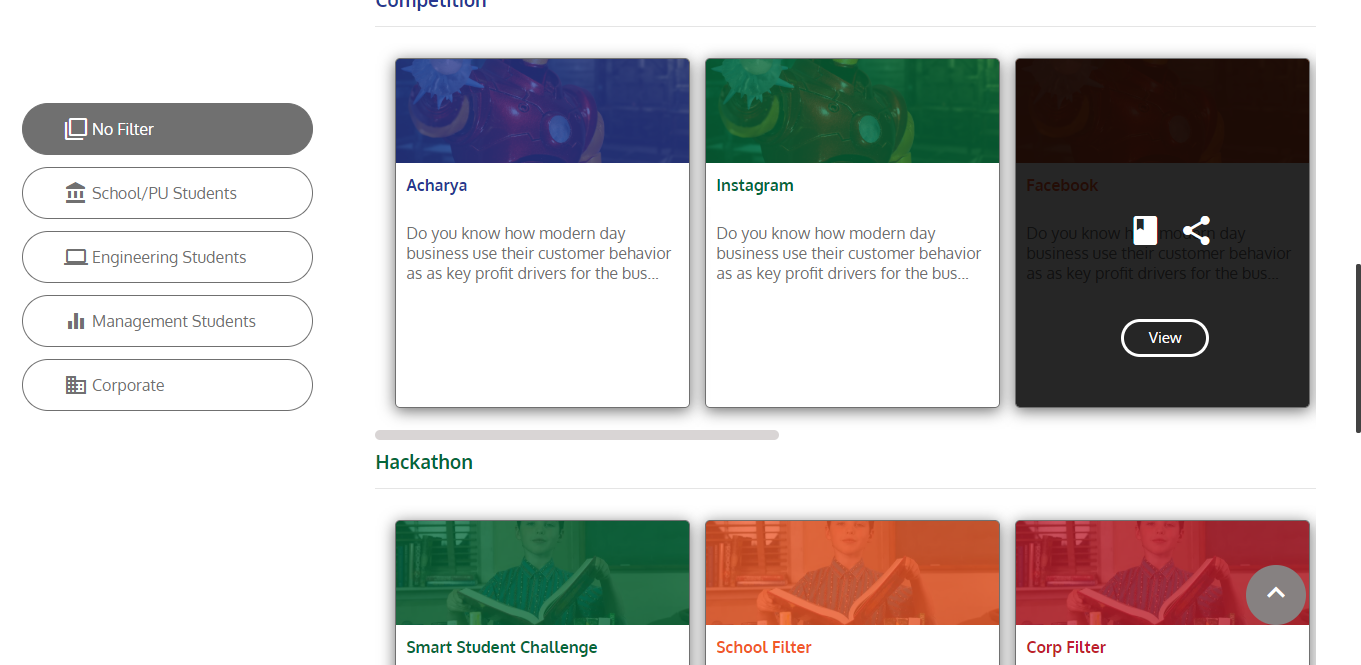
****

Fig 5.1(b): Home page of the Product.

Chapter 6

learning outcomes

# **Chapter 6**

# **learning outcomes**

After successful completion of Internship, I learned the following.

* Create device-independent Angular applications (web-app).
* Transferring data from/to any Angular based project.
* Routing through the Angular system.
* Develop Components using TypeScript, Templates, and Decorators.
* Consume REST services using Observables.
* Modularize applications with the Component Router.
* Capture and validate input with template-driven forms.

**References:**

### https://getbootstrap.com/docs/4.1/getting-started/docs/

### https://www.udemy.com/the-complete-guide-to-angular-2/

### https://angular.io/docs

stipend

# **stipend received details**

As the Internship took about three months form 10th July to 14st August, I was provided with a stipend of 5,000/- Rs per month.