**CHAPTER 1**

**INTRODUCTION**

**1.1 OVERVIEW**

Role of Artificial intelligence plays major role many fields especially in the areas where the human intervention or human error is to be avoided. This need not even left the gaming field also ,the reason is people are tired of playing with traditional computer opponent so they wanted to play with the human opponent so there came the need for multiplayer online and offline games ,But later the online multiplayer games where over flown with online players. This too didn't last for long because of the players busy schedule players who want to play in their free time didn't find the other players online ,so there came a need for the AI to act in the place of human player. This project is a multi player snake game in which two player compete each other on a single board.

This AI substitute of the human player should allow the opponent players to win as well as to loss . So the AI should be build in such a way that it plays like a actual human player. This project has developed a AI to capture the food and to escape from the opponent's attacking moves and patterns he follows to defeat his opponent.

The bigger challenge in this project is the data collection from the user and facilitating the players to player with their friend ,So thinking of the better way to connect the people the first thing that comes to our mind is the web technologies which has played and being playing a major role many field like banking ,entertainment ,medicine and education. So a better platform to implement this idea is the web technologies

Now the disclosed information is that the project is entire depend on web technology ,but how the a web technology is going to serve a need of the gaming project because the gaming need a lots of data to be transferred over the network for smooth and clean gaming experience with the friends and strangers.

This need is satisfied by the many server push mechanisms ,one such mechanism formulised is SSE(Server Sent Events). The next challenge is the data storage and collection .The better way to store the data in a web environment is the database ,this project uses the MySQL database to store the data and also for retrieval ,manipulation and updation of data.

**1.2 MULTIPLAYER SNAKE GAME**

This Snake game is mainly focused on the multiplayer mode rather than the single player mode. The major challenge of the multiplayer mode over single player mode is the end point of the game .In single player mode the end point of the game is none ,it depend on the player's ability and only the score matters but in multiplayer mode the end point is entirely different ,any one of the snake should stand or both should fall so the end point is led by attacking each other ,even if they don't attack the game board cannot accommodate their body after they eat several food. In this multiplayer concept the player can attack each other ,if one player attack the other at particular body part then the parts after that will be discarded ,in this way their body part will be reduced and the one with body part less than the desired level will be declared as lost and the other as winner. Here a doubt may arise, then how will they develop their body part .This operation is done by consuming the food displayed in the board. This food is also distributed in synchronised manner that is both the players will get same food at same co-ordinates and there is no different weightage for different food, every food will have same weightage .

**1.3 CHARACTERISTICS OF THE MULTIPLAYER SNAKEGAME**

Received moves as result of the several games played by the player among their friends more and more amount data were got accumulated in the database which could be properly maintained and manipulated to give excellent user experience .This data is collected while the player gives any moves to their snake in any given situation .Thus the collected data is used by the computer during the game play with the other human player ,for example consider both the player and the computer are approaching for a food now the human player will have clear idea of what he/she should do but a computer will not have such an idea or do not have any prediction of the opponent’s next move ,so to make the computer comfortable with the opponent human player these data gives wide view for the computer. Some more characteristics that are added are to this game is that the computer snake reveals its current state that is it change its snake colour when it can predict the opponent’s next move and change to another colour when it cannot predict the opponent’s next move. The reason to have this feature is to increase the curiosity of the opponent player so that his interest in the game increases and he/she give their valuable move data to computer for its future use.

This game follows the steps below:

1- Collect the data from the user by a multiplayer game session.

2- Do some prediction job when a situation is met .

3- Uses the predicted data to escape from the opponent.

**1.4 SERVER PUSH MECHANISM FOR MULTIPLAYER GAME**

The first model proposed is entirely a server oriented model in which the entire game data is maintained in the server side and continuously sent as a event stream to the client, the major drop back of this model is the efficiency in term of both time and space occupied by the data transferred over the network. The reason for this failure is due to the enormous amount of object created by the both client and server. This failure is overcome by the second model which completely localized the to the individual client machines by the client agent like browser. This model has improved the efficiency of the client as well as the server which should serve the several similar clients. So here come the need for the synchronization because the game is left to the individual client and only some data are sent to the server regarding the multiplayer session.

**1.5 PREDICTION FOR BOT**

In this project the prediction is done to guess the human opponent’s next move and make a relevant action to the computer snake. This prediction is basically done by calculating the likelihood of any given classes. This game involve totally four classes they are left ,right ,top and down .These four classes are classified based on eight parameters they are head position of the snakes and distance between food and snake and length and current direction of the snake.

**1.5.1 Database for Prediction Model**

**Moves database**

This is the database which contains various moves made by the player in their previous games, the attributes of the subject relation are headx1,heady1,l1,d1,id1,id2,d2,l2,dixs,diyx,dix1f,diy1f and game board id.

**CHAPTER 2**

**LITERATURE SURVEY**

|  |  |
| --- | --- |
| **2.1** | **A MACHINE LEARNING APPROACH FOR TRACKING AND PREDICTING STUDENT PERFORMANCE IN DEGREE PROGRAMS** |

Accurately predicting students' future performance based on their ongoing academic records is crucial for effectively carrying out necessary pedagogical interventions to ensure students' on-time and satisfactory graduation. Although there is a rich literature on predicting student performance when solving problems or studying for courses using data-driven approaches, predicting student performance in completing degrees (e.g., college programs) is much less studied and faces new challenges: (1) Students differ tremendously in terms of backgrounds and selected courses; (2) courses are not equally informative for making accurate predictions; and (3) students' evolving progress needs to be incorporated into the prediction. In this paper, they deal with developing a novel machine learning method for predicting student performance in degree programs that is able to address these key challenges. The proposed method has two major features. First, a bilayered structure comprising multiple base predictors and a cascade of ensemble predictors is developed for making predictions based on students' evolving performance states. Second, a data-driven approach based on latent factor

models and probabilistic matrix factorization is proposed to discover course relevance, which is important for constructing efficient base predictors.

**2.1.1 Applications**

* Novel machine learning method to predict student performance.
* Difference in selected courses, courses not equally informative, progress of students not tracked.
* Data driven approach to find course relevance – course clustering – only relevant inputs given.
* Handle heterogeneous data – discover correlation among courses.
* Base layer and ensemble predictor layer for evolving progress.

**2.1.2 Advantages and Disadvantages**

* Valuable information to recommend courses, impact curriculum design.
* Elective courses not considered.

|  |  |
| --- | --- |
| **2.2** | **PREDICTING STUDENTS’ GPA AND DEVELOPING INTERVENTION STRATEGIES BASED ON SELF-REGULATORY LEARNING BEHAVIORS** |

Predicting students' grades has emerged as a major area of investigation in education due to the desire to identify the underlying factors that influence academic performance. Because of limited success in predicting the grade point average (GPA), most of the prior research has focused on predicting grades in a specific set of classes based on students' prior performances. The

issues associated with data-driven models of GPA prediction are further amplified by a small sample size and a relatively large dimensionality of observations in an experiment. In this paper, they have utilized the state-of-the-art machine learning techniques to construct and validate a predictive model of GPA solely based on a set of self-regulatory learning behaviors determined in a relatively small-sample experiment. They have quantified the predictability of each constituents of the constructed model and discussed its relevance. Ultimately, the goal of grade prediction in similar experiments is to use the constructed models for the design of intervention strategies aimed at helping students at risk of academic failure. In this regard, they lay the mathematical groundwork for defining and detecting probably helpful interventions using a probabilistic predictive model of GPA. They demonstrate the application of this framework by defining basic interventions and detecting those interventions that are probably helpful to students with a low GPA. The use of self-regulatory behaviors is warranted, because the proposed interventions can be easily practiced by students.

**2.2.1 Applications**

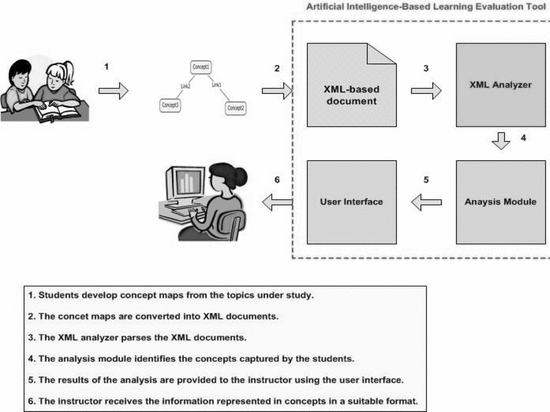
* Validate predictive model based on set of behaviors.
* Use constructed models to design intervention strategies.
* Study of time management, concentration, elaboration.
* Maximum weight first order dependence tree (MWDT)
* Naïve Bayes for classification
* Discriminate low and high GPA – cross validation

**2.2.2 Advantages and Disadvantages**

* Can translate MWDT to intervention strategy.
* Cannot predict usage of time, different types of questions not considered

|  |  |
| --- | --- |
| **2.3** | **ARTIFICIAL INTELLIGENCE-BASED STUDENT LEARNING EVALUATION: A CONCEPT MAP-BASED APPROACH FOR ANALYZING A STUDENT'S UNDERSTANDING OF A TOPIC** |

In this paper, they describe a tool coined as artificial intelligence-based student learning evaluation tool (AISLE). The main purpose of this tool is to improve the use of artificial intelligence techniques in evaluating a student's understanding of a particular topic of study using concept maps. Here, they calculate the probability distribution of the concepts identified in the concept map developed by the student. The evaluation of a student's understanding of the topic is assessed by analyzing the curve of the graph generated by this tool. This technique makes extensive use of XML parsing to perform the required evaluation. The tool was successfully tested with students from two undergraduate courses and the results of testing are described in this paper.



**Figure 2.1 Artificial Intelligence-Based Learning Evaluation Tool**

**2.3.1 Applications**

* Concept map – visual representation of a topic – knowledge retention – evaluation of learning.
* Understand student psychology – practical applications.
* AISLE – concept map to XML – parse and analyze concepts – results
* given to teacher – suitable form
* Concepts have unique ID – link phrases
* Hierarchy of interlinked concepts
* Score – concepts and levels

**2.3.2 Advantages and Disadvantages**

* Compare maps, validate testing and grading
* Developed from scratch, not accurate compared to other methods.

|  |  |
| --- | --- |
| **2.4** | **COMPUTERIZED ADAPTIVE TESTING BASED ON DECISION TREE** |

Since the early days of Computerized Adaptive Testing (CAT), the maximum information method based on Item Response Theory (IRT) has been commonly used. This method selects the optimal item that maximizes the test information (Fischer information measure) at the current estimated ability based on IRT from an item bank. Another famous item selection method is Owen's Bayesian method. His method is an empirical or sequential Bayes procedure where at each stage the previous posterior distribution of the unknown parameter serves as the new prior distribution. Owen's method provides less accuracy estimation of examinee's ability than the method using Fischer's information because Owen's method ignores item discrimination parameters in IRT. This paper proposes a new computerized adaptive testing

employing a decision tree model, instead of test theories. The attribute variable of the model is examinees' responses to each item and the output variable is examinees' test total scores. Some simulation experiments show better performances of the proposed method compared to the traditional methods and solve the problems.

The advantages of the proposals are as follows:

1. This method has less item selection bias from an item bank than Fischer information based methods.

2. This method has no estimation error in the beginning of the test.

3. This method assumes a network probabilistic structure of items instead of a simple local independent structure.

4. Item selection according to a learned tree reduces computational costs compared to the traditional methods.

Furthermore, this paper provides some simulation experiments to compare the proposed method with the performances of the traditional item selection methods. As the results, the proposed method shows the best performances.

|  |  |
| --- | --- |
| **2.5** | **DESIGN AND APPLICATION OF A RESPONSE-DRIVEN ADAPTIVE TEST SYSTEM** |

Unlike the traditional paper-and-pencil tests, computer adaptive testing (CAT) presents the next item based on his or her past responses, which is more appropriate for a particular individual. However, it is rarely seen in a smaller-scale scenario, such as in classrooms or business because of the complexity of Item Response Theory (IRT) and the possible usages of services may not be known or may increase due to the development of new services. The approach to this problem is self-adaptive functional testing, which is a combination of monitoring and active testing at runtime. They present a Response-Driven Adaptive Test System (RDATS) to get user’s response data which has been detected during monitoring, existing test cases are adapted automatically to the specific situation and applied afterwards based on user’s response at runtime. In this paper, they discuss the structure and functionality of the system with particular emphasis on adaptive test Mechanisms and SQL database design.

**CHAPTER 3**

**PROPOSED WORK**

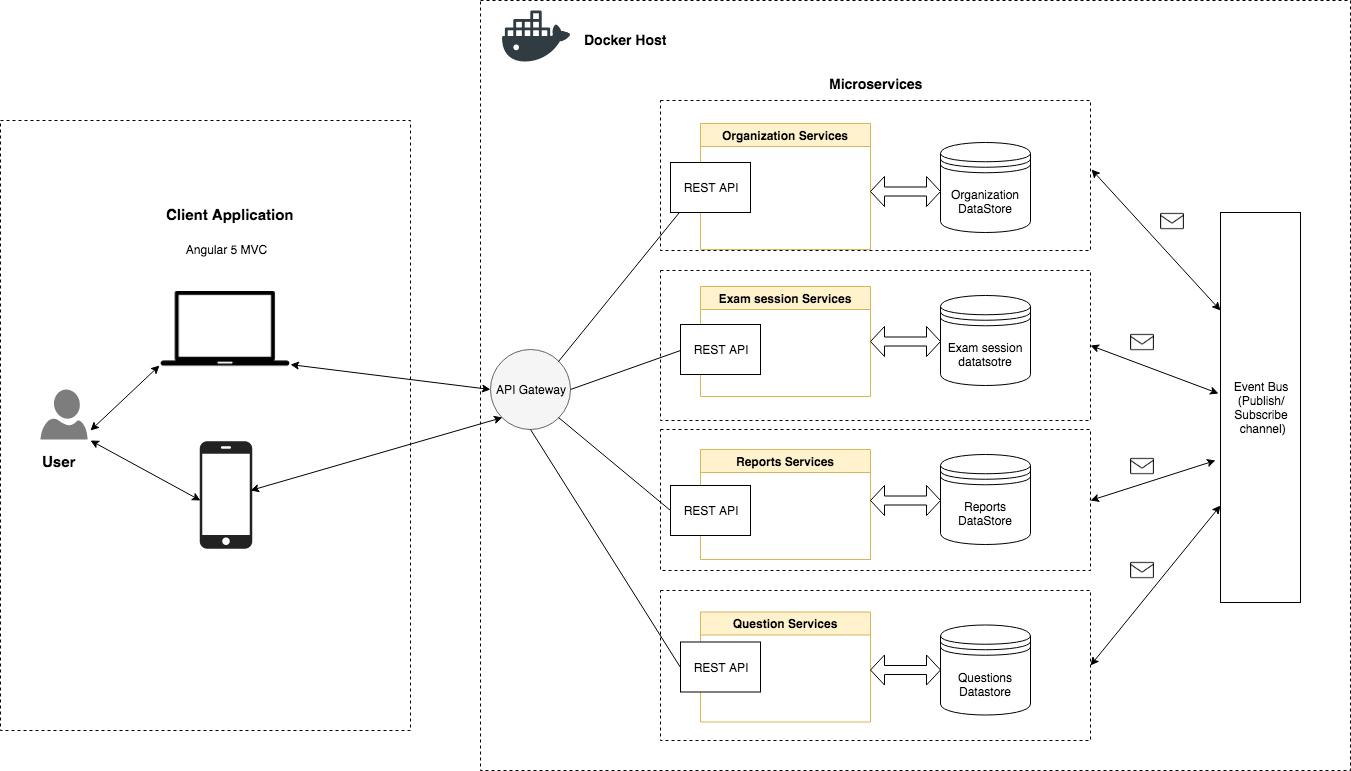
**3.1 RECOMMENDED METHOD**

Microservices - also known as the microservice architecture - is an architectural style that structures an application as a collection of loosely coupled services, which implement business capabilities. The microservice architecture enables the continuous delivery/deployment of large, complex applications. It also enables an organization to evolve its technology stack.

**3.1.1 Advantages**

* Service enabled, independently running components.
* Evolutionary design for getting the best possible decomposed system, which can be replaced and upgraded without affecting its collaborator.
* The central idea behind microservices is that our exam portal becomes easier to build and maintain when they are broken down into smaller, composable pieces which work together.
* Each component is continuously developed and separately maintained, and the application is then simply the sum of its constituent components. This is in contrast to a traditional, "monolithic" application which is all developed all in one piece.
* Applications built as a set of modular components are easier to understand, easier to test, and most importantly easier to maintain over the life of the application.
* It enables us to achieve much higher agility and be able to vastly improve the time it takes to get working improvements to production. This approach has proven to be superior, especially for large enterprise applications which are developed by teams of geographically and culturally diverse developers.

**3.2 PROPOSED ARCHITECTURE DIAGRAM**

****

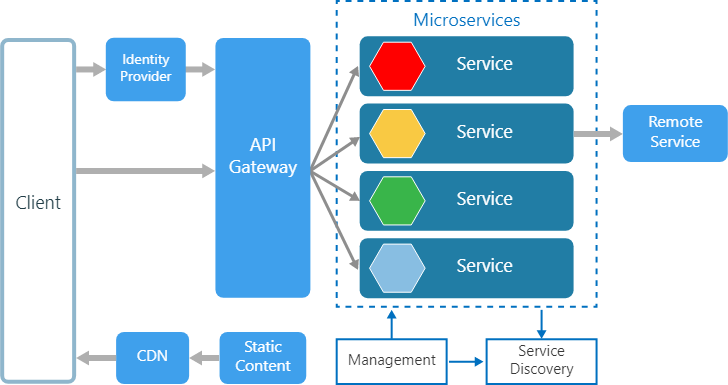
**Figure 3.1 Architecture Diagram**

**3.3 DESCRIPTION OF THE ARCHITECTURE DIAGRAM**

**3.3.1 Microservices**

Microservices is a variant of the [service-oriented architecture](https://en.wikipedia.org/wiki/Service-oriented_architecture) (SOA) architectural style that structures an [application](https://en.wikipedia.org/wiki/Application_(computing)) as a collection of [loosely coupled](https://en.wikipedia.org/wiki/Coupling_(computer_programming)) services. In a microservices architecture, services should be [fine-grained](https://en.wikipedia.org/wiki/Service_granularity_principle) and the [protocols](https://en.wikipedia.org/wiki/Protocol_(computing)) should be lightweight. The benefit of decomposing an application into different smaller services is that it improves [modularity](https://en.wikipedia.org/wiki/Modular_programming) and makes the application easier to understand, develop and test. It also parallelizes [development](https://en.wikipedia.org/wiki/Software_development) by enabling small autonomous teams to develop, [deploy](https://en.wikipedia.org/wiki/Software_deployment) and scale their respective services independently. It also allows the architecture of an individual service to emerge through continuous [refactoring](https://en.wikipedia.org/wiki/Refactoring).

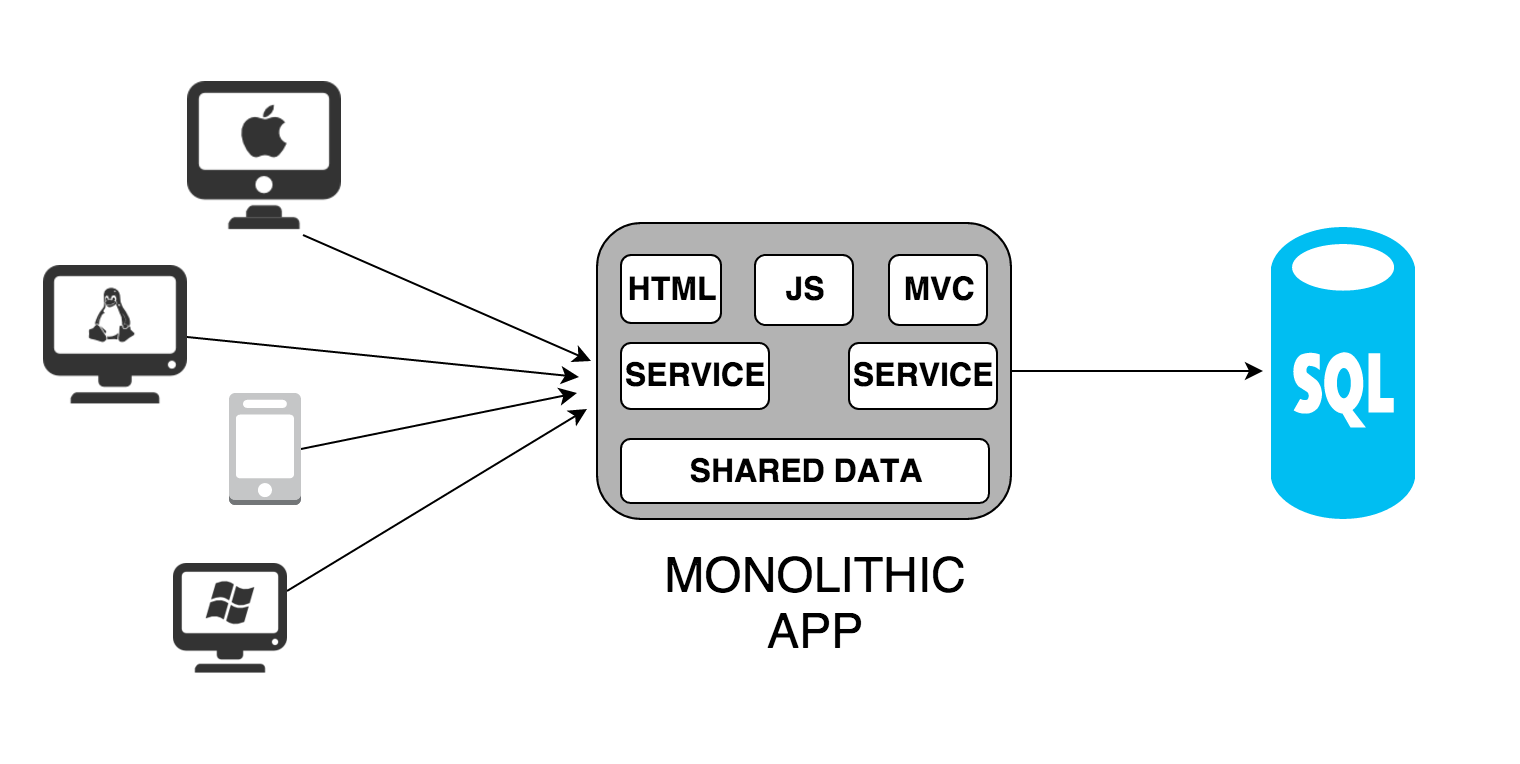
In monolithic architectural design we create a big cumbersome application with all modules tightly coupled inside a single executable, which is typically deployed on a web or application server whereas, Microservices architectural style defines a setup, where application components are standalone applications of their own. These independent application components talk to each other either using RMI (Remote Method Invocation), Restful Web Services or Push Messaging. In a monolithic service oriented architecture deployment, each small change meant that the entire monolith needed to be rebuilt and this, in turn, meant that rebuilds weren't happening as rapidly as they should.



**Figure 3.2 Microservice Architecture**

When everything is an API, communication between application components becomes standardized. All a component has to do to make use of your application and data is to be able to authenticate and communicate across those standard APIs.

A typical monolithic architecture application looks like:



**Figure 3.3 Monolithic Architecture**

There are some disadvantages to this architectural design. These disadvantages or drawbacks have become the strengths of microservices architecture:

1. No frequent and easy releases - As monolithic applications grow in size, due to tight coupling between components, it becomes difficult to do easy and frequent releases. Release planning takes a lot of time of people from various groups. Frequent release is discouraged for making sure the application should not break due to the newly released feature.
2. Problem in continuous delivery - We may not notice this problem if the application is small. In case of bigger monolithic applications, deployment times can be frustratingly long and slow. If a single change to the application would require the entire application to be redeployed, then this could become an obstacle to frequent deployments, and thus

an impediment to continuous delivery. This could be a serious issue if you are serving a mobile application where users expect the latest cool new features all the time.

1. Difficult to manage team and project - Project management has its own challenges in monolithic application development. Even a modularized application has interdependency in terms of deployment and release. It takes a toll in terms of time and effort to plan the release and manage tightly coupled interdependent modular development.
2. Expensive scalability & performance - It is possible to scale a monolithic application but the cost is very high.
3. Lack of technology diversity - When we choose a technology stack for a monolithic application, we consider a balanced stack which can serve well for all of our requirements. We can not employ specific technology for specialized needs.
4. Not easy to replace components - It is very difficult to replace any component with a better design or performance without affecting the whole architecture.

**3.3.2 REST API**

REST stands for Representational State Transfer. REST is a web standards based architecture and uses HTTP Protocol for data communication. It revolves around resources where every component is a resource and a

resource is accessed by a common interface using HTTP standard methods.

In REST architecture, a REST Server simply provides access to resources and the REST client accesses and presents the resources. Here each resource is identified by URIs/ Global IDs. REST uses various representations to represent a resource like Text, JSON and XML. JSON is now the most popular format being used in Web Services.

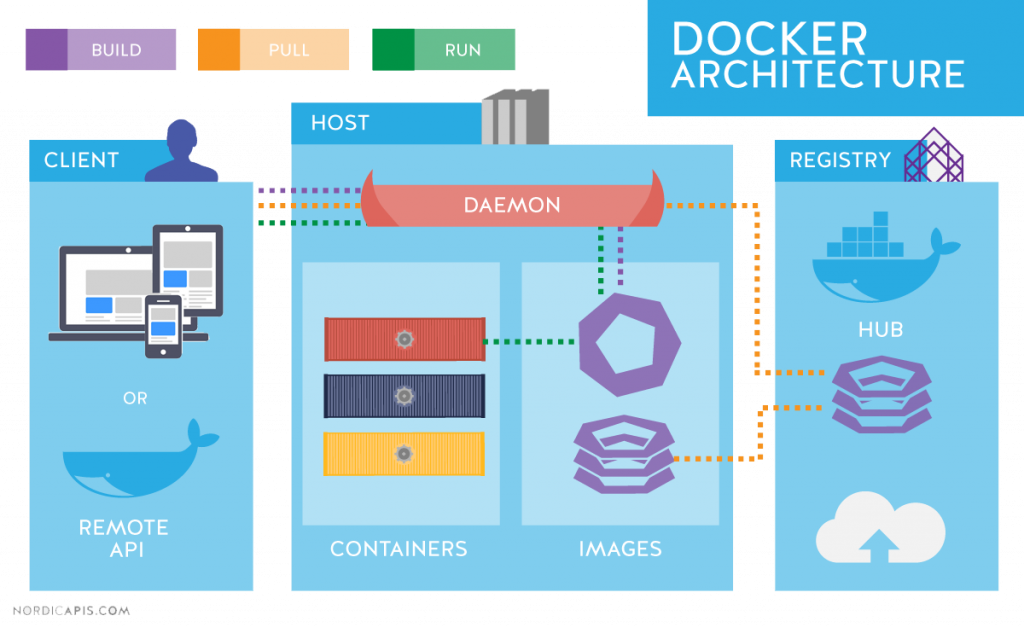
The REST API gateway definitely adds value - when the application is split into smaller units (the microservices) - it allows to easily manage, monitor and govern the API interfaces centrally as a single unit, separately from the implementation(s).

**3.3.3 Using Docker for Microservice**

One issue developers have noticed with microservices is with accessibility and frequency. If the user accesses the microservices often, the response time and productivity slows in [virtual machines](http://searchservervirtualization.techtarget.com/definition/virtual-machine). Containerization, using to run and deploy applications without launching an entire VM for each application, allows microservices to move more quickly. Docker lets us deploy microservices one by one on a single host (server).

Each microservice can be further broken down into processes running in separate Docker containers, which can be specified with Dockerfiles and Docker Compose configuration files. Combined with a provisioning tool, each

microservice can then be easily deployed, scaled, and collaborated on by a developer team. Specifying an environment in this way also makes it easy to link microservices together to form a larger application. The Docker technology of the container, now emulated by other container services, helps address the biggest challenges to building a microservice architecture in the following ways.

****

**Figure 3.4 Docker Architecture**

**Task isolation**

Create a [Docker container](https://www.sumologic.com/devops/design-microservices-architecture-with-docker-containers/docker-containers-management/) for each individual microservice. This solves the problem of resource bloat from over provisioned instances idling under the almost non-existent strain of a lone service, and multiple containers can be run per instance.

**Support multiple coding languages**

Each service can be run in a different language, including libraries and framework information, into linked containers to simplify and manage multiple platforms

### **Database separation**

Use containers to host one or more data volumes, then reference them from other microservices and containers.

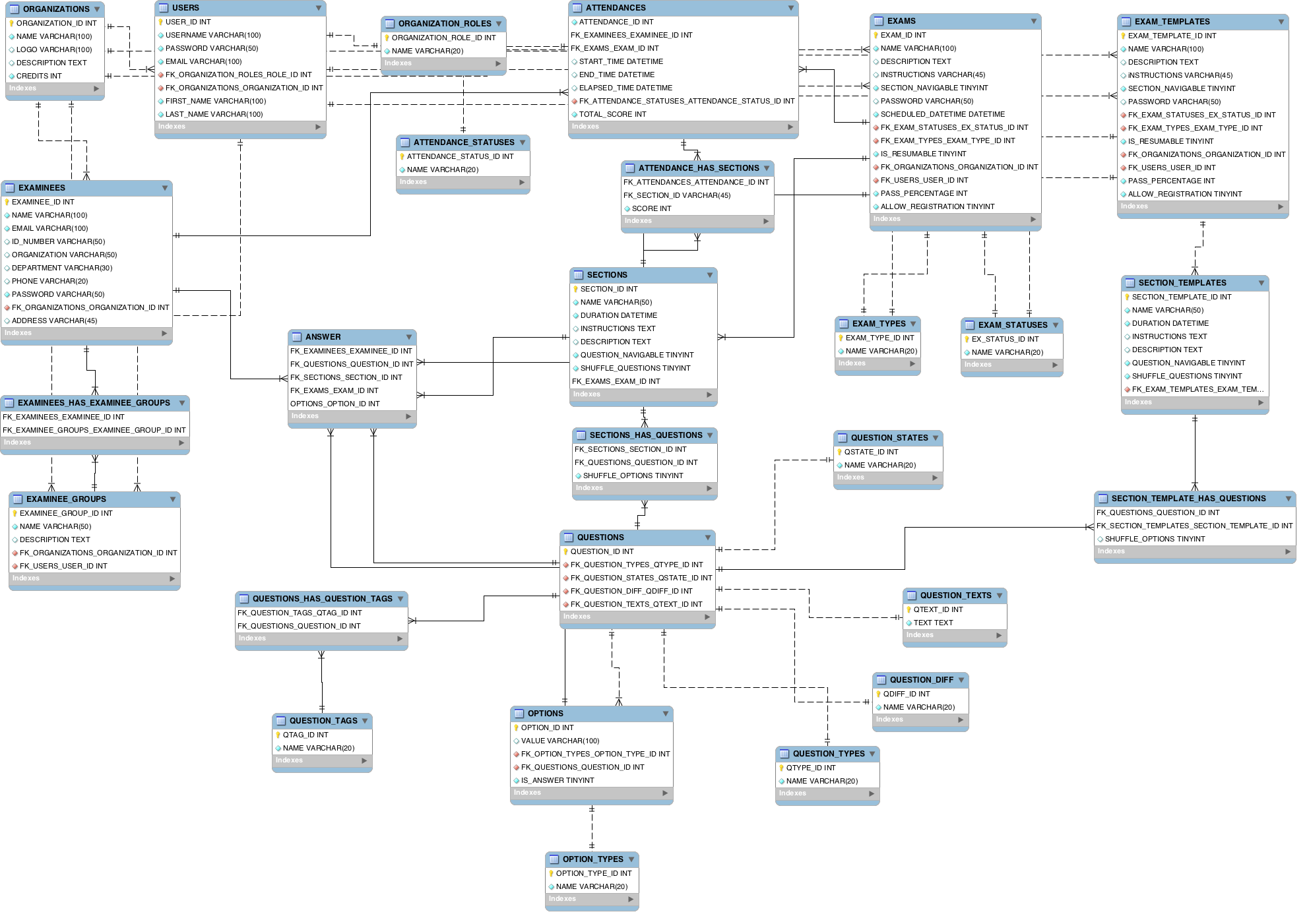
**Infrastructure Optimization**

Docker has much less overhead than a full-fledged Virtual Machine. Though Docker can be deployed on bare metal with Linux OS, many of the consumers already have adopted Virtualization technologies and already have Virtual Machines for deploying the applications. Docker containers complement Virtual Machines technology and can even be hosted inside Virtual Machines. But using Docker containers, one can create an optimized infrastructure with fewer VMs to manage, with more Docker containers for every VM. Because Docker is based on the design of the Linux container, the same operating system on the VM can be shared by the Docker containers thereby increasing the density of the infrastructure.

**3.3.4 Publish–subscribe pattern**

Publish/subscribe messaging, or pub/sub messaging, is a form of asynchronous service-to-service communication used in server less and microservices architectures. In a pub/sub model, any message published to a topic is immediately received by all of the subscribers to the topic. Pub/sub messaging can be used to enable event-driven architectures, or to decouple applications in order to increase performance, reliability and scalability.

**3.4 DATABASE SCHEMA**

****

**Figure 3.5 Database Schema**

**CHAPTER 4**

**SYSTEM REQUIREMENT**

**4.1 HARDWARE REQUIREMENTS**

PROCESSOR : Intel Core i3 and above

HARD DISK DRIVE : 50GB+

RAM : 1GB+

**4.2 SOFTWARE REQUIREMENTS**

OPERATING SYSTEM : Linux - Ubuntu 14.0.4

FRONT END : HTML5, CSS3, Angular 5

BACK END : Node.js 8.10, Java 8, PHP 7.0

SESSION STORAGE : Apache CouchDB 2.2.1

DATABASE : MySQL 5.7

RUNTIME PLATFORM : Docker

**CHAPTER 5**

**IMPLEMENTATION MODULES**

**5.1 LOGIN SERVICE**

**5.1.1 Description**

One of the main arguments of introducing an API Gateway is handling cross cutting concerns like authentication, authorization and security which are handled using JSON Web Token (JWT) tokens.

In the web-based exam system, the user name and the password are the keys of the system to identify a specific user. Since any person with the user name and the password is considered to be authorized to access into the system, the information security is critical. The system can automatically generate a token for students registered and send this information to parties in a secure way.

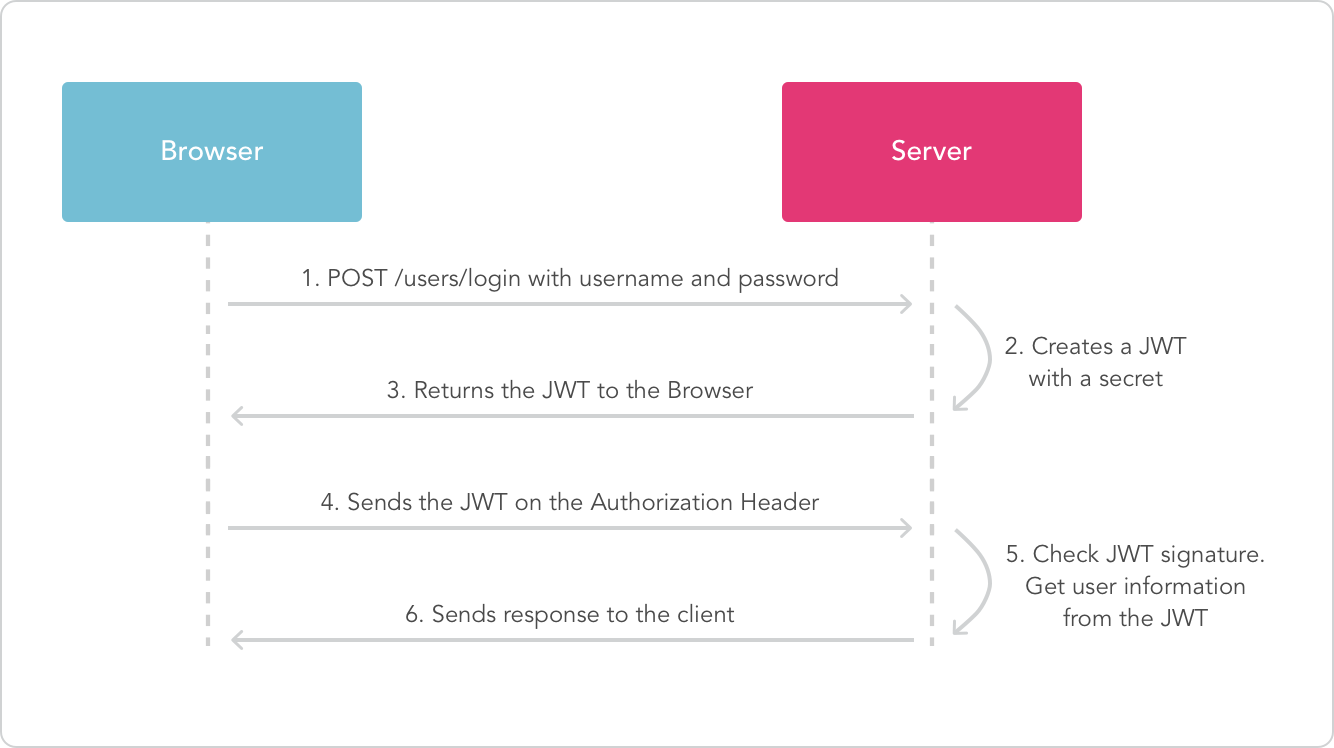
JWT authentication process can be broken into following four steps-

1) User is validated against database and claims are generated based on user’s role.

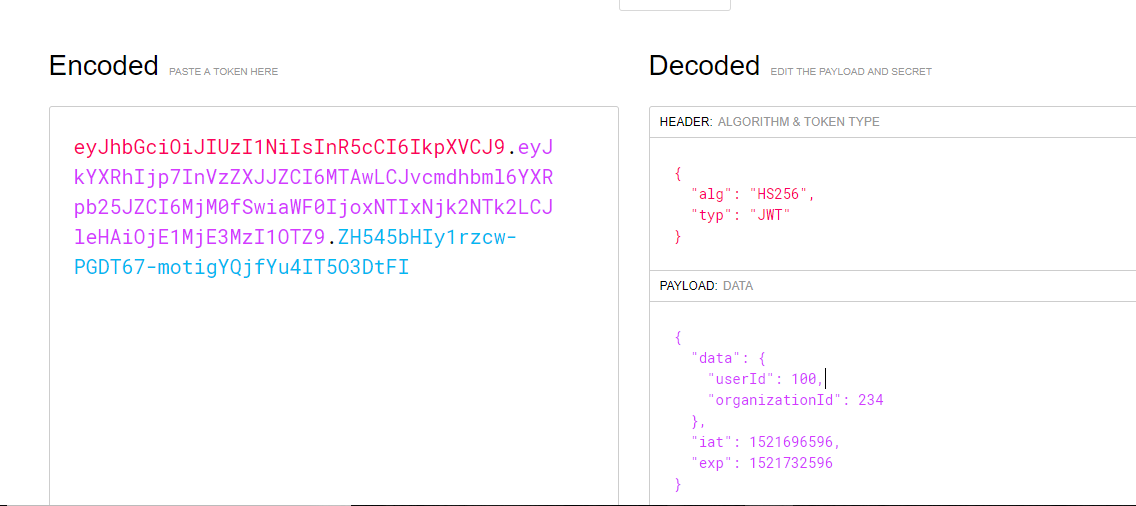
2) Payload containing claims or other user related data is signed with key to generate token and passed back to user.

3) User sends this token with each request, normally in header or cookies and then received token is decrypted to validate claim.

4) Once user is identified, User is allowed to access Resource server based on his claim.



**Figure 5.1 Authentication using JWT token**



**Figure 5.2 Encoding the Payload**

**Generating JWT Token**

// if user is found and password is right  
 // create a token with only given payload  
 // we don't want to pass in the entire user since that has the password  
 const payload = {  
 admin: user.admin

};  
 var token = jwt.sign(payload, app.get('superSecret'), {  
 expiresInMinutes: 1440 // expires in 24 hours  
 });  
  
 // return the information including token as JSON  
 res.json({  
 success: true,  
 token: token  
 });

**Verifying JWT Token**

// check header or url parameters or post parameters for token  
 var token = req.body.token ;  
 // decode token  
 if (token) {  
 // verifies secret and checks exp  
 jwt.verify(token, app.get('superSecret'), function(err, decoded) {   
 if (err) {  
 return res.json({ success: false, message: 'Failed to authenticate token.' });   
 } else {  
 // if everything is good, save to request for use in other routes  
 req.decoded = decoded;   
 next();  
 }  
 });  
 } else {

// if there is no token, return an error  
 return res.status(403).send({   
 success: false,   
 message: 'No token provided.'   
 });

}  
 });

**5.2 ORGANIZATION SERVICE**

**5.2.1 Description**

There are three types of users in the system: System administrator, Super Admin of the organization registered and other users of the organization. System administrator has the authorization to add super admins.

Super admins are authorized to define other users of the organizations. Super admins and users of the organization can add and modify the questions, change student passwords, edit questions, access exam results, transform exam results into the excel form for further analysis, and send messages to parties. When students access into the system, they see the exams they are required to take; they can take these exams; and they can view their exam scores.

**5.3 QUESTION SERVICE**

**5.3.1 Description**

This section deals with the way the questions are selected for the exam. Along with that it maintains a questions bank where addition, updation and deletion of questions can be performed.

There are three kinds of systems for paper generation:

(1) the database will be viewed as the questions set for selection by examiners

(2) the test paper set will be used for random selection;

(3) automatic paper generation will be implemented using an adaptive strategy.

The first and second systems are able to guarantee quality. But the first one works similarly to manual operation, while the second limits the number of papers. The third type uses random selection and could form thousands of papers almost without limitation. Under the control of the state space, a random selection method may be proved to carry out a large number of invalid operations in the dead cycle, which eventually leads to the failure of the examination papers. The adaptive genetic algorithm has intelligent search technology with global optimization. Good convergence characteristics can work well for meeting the requirements for automatic test papers. Along with automatic questions selection, the organizations can have their own question bank. The adaptive algorithm can pick questions only form the questions uploaded by the organization. The organizations can have their questions as public to all organizations and private only to their organizations. They can add or delete or update the questions at any point of time.

function authenticate(request) {

token = getTokenFromHeader();

if(token) {

userDetails=getUserIdFromToken(token);

if(userDetails === -1) {

// unauthorised user

return 401;

}

// valid user

return true;

}

else {

// bad request

return 400;

}

}

function getQuestions(request) {

authenticateResult = authenticate(request);

if( authenticateResult == true) {

fetch questions from database;

apply the selected filters;

return the questions as response;

} else if (authenticateResult == 401) {

return 401-response;

} else if (authenticationResult == 400) {

return bad-request-response;

}

}

**5.4 EXAM SERVICE**

**5.4.1 Description**

Exam session service deals with the entire flow of the examiner creating an exam. It includes creating an exam template and adding questions.

Exams can be created by two different methods.

In the first method, examiner can create an exam by selecting as many as questions they need with a certain difficulty level from the question pool that they previously created by entering questions with different difficulty levels. While the exam is being created, which group to take it is predetermined. Thus, students view the exam created for them when they access into the system. On the other hand, in the second method, the lecturer lists all questions recorded in the question pool for the relevant exam, then selects the desired questions among them.

Exam Template can be created by two different methods.

In the first method, creating a new template which includes the examiner defining the name of the exam, creating password,scheduling date and time, defining the navigability of the questions. While the exam is being created, which group to take may or may not be predetermined. Thus, students view the exam created for them when they access into the system

On the other hand, in the second method, the examiner reuses the previously created template and making changes.

**5.5 EXAM SESSION SERVICE**

**5.5.1 Description**

Exam session service deals with the entire flow of the user taking up the examination. It involves from the time, the user logs in to take up the exam until the time the user submits the examination. Though the flow looks simple, this service is crucial in terms of handling dynamic data and security issues. Various security issues handled in this service are as follows:

* Crashing of system during the duration of the examination.
* Resuming the examination session on the approval of the organizer due to the external issues faced by the examinee.
* Preventing multiple exam-session for the same examinee.
* Blocking the user from changing the time duration of the exam.
* Avoiding question leakage to the user.
* Maintaining the time duration of the exam session in the server side.
* Contacting the timer in the server every second to be in sync with the front end.

These issues being crucial for any examination, are handled in such a way that the user cannot perform any kind of malpractice during the exam-session.

Apache CouchDB is used to store the exam session object for the user once the user begins the exam. As an extra security feature, at one particular time, only one session object can be created for an examinee. Apache CouchDB stores the data safely, on our own server, or with any of our leading cloud provider. Our application interacts with CouchDB, as it speaks JSON natively and supports binary for all our data storage needs. CouchDB comes with a developer-friendly query language, and MapReduce for simple, efficient, and comprehensive data retrieval.

**5.6 REPORT SERVICE**

**5.6.1 Description**

Report service deals with the analysis of the exams conducted by the organization. This service does the job of correction of each and every individual examinee and stores their score in the database server. The above process is performed only after the examinee submits their exam. The report service shows reports for multiple exams which all got over. They show details attendance report, section wise topper overall topper, which question has the highest and lowest success rate , they can sort the details in any order they want. The report service gives information of each and every individual and shows their strong and weak zones based on their marks. As a result no deserving candidate is left.

The reports can be displayed in tabular as well as graphical format. Server Side Pagination is used for displaying details in front end. The reports are the most important thing and organizations infer many things from report and take decisions based on it so it must need the highest level of security. No one has the authority to change the answers of an examinee after submitting the exam. The reports can be bookmarked, shared with other persons of same organization. The report service gives the end organization multiple ways to share the reports. It can be downloaded in multiple formats like excel sheets, pdf, documents, etc. Images of the graphs can be downloaded in high resolution. The organization can also set passwords for these files an additional level of security.

**CHAPTER 6**

**SNAPSHOT OF MODULES**

**6.1 EXAM LOGIN PAGE**

****

**Figure 6.1 Exam Login Page**

**6.2 EXAMINEE LOGIN PAGE**



**Figure 6.2 Examinee Login Page**

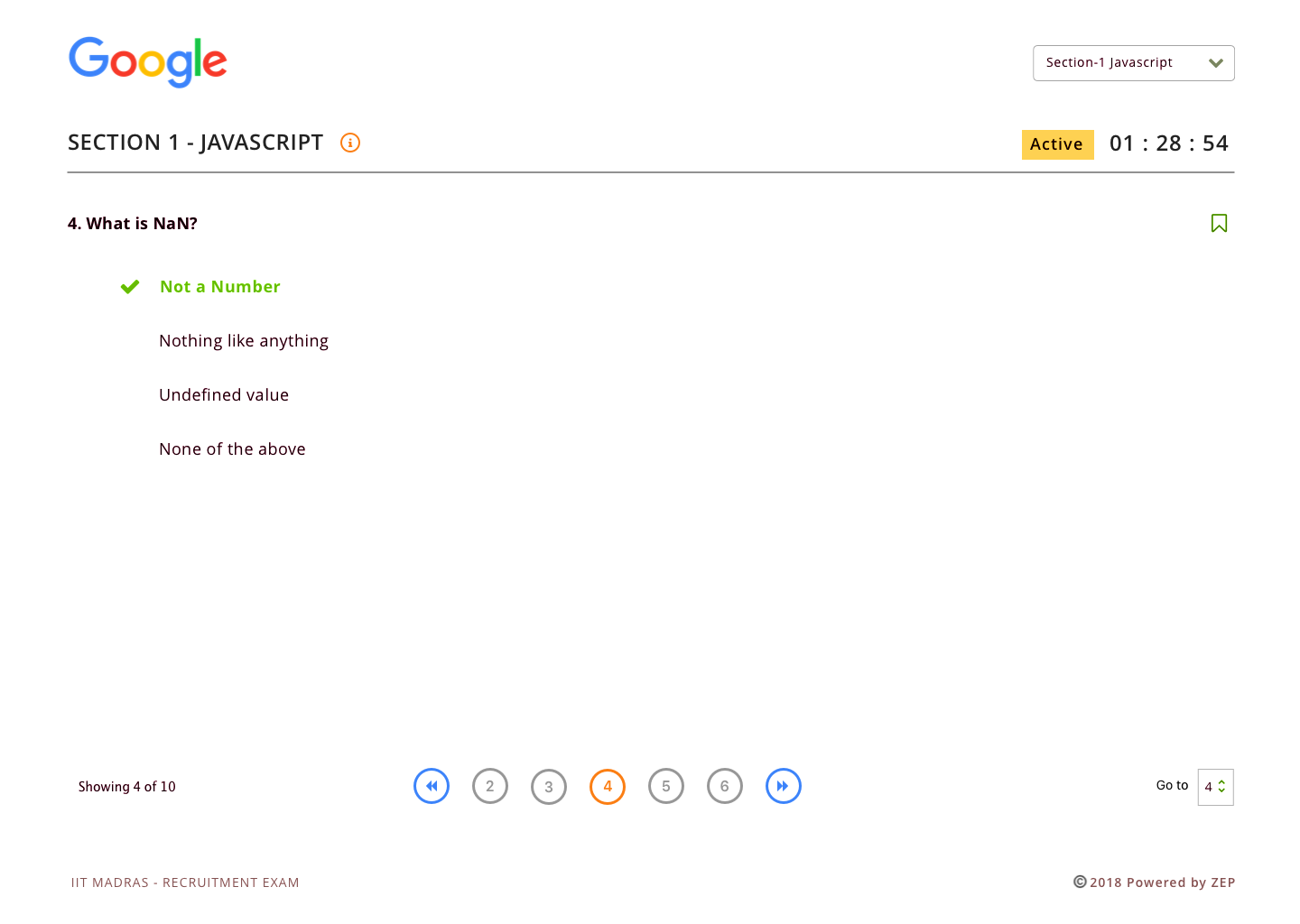
**6.3 EXAMINEE REGISTRATION PAGE**



**Figure 6.3 Examinee Registration Page**

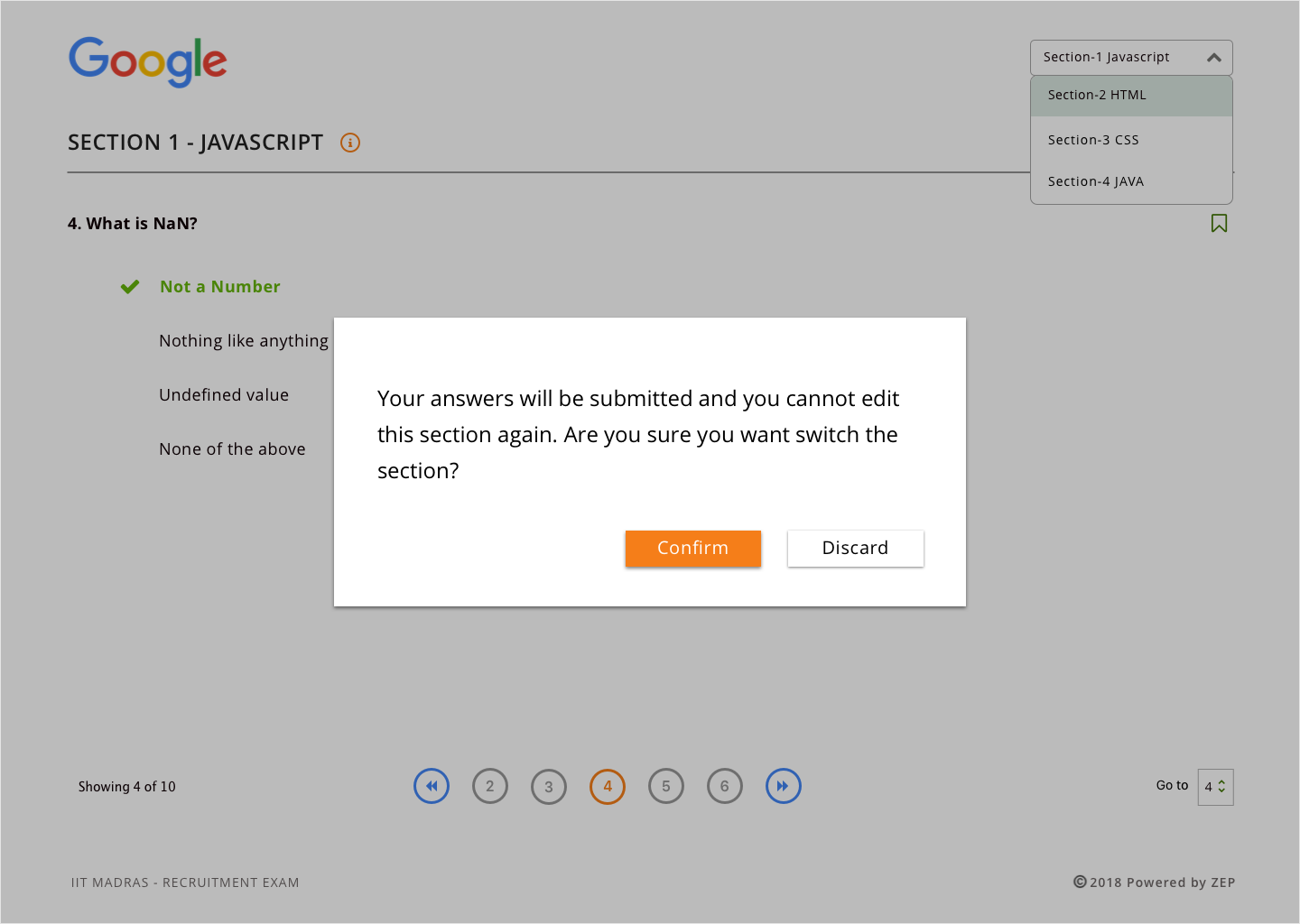
**6.4 EXAM SCREENS**

**6.4.1 Exam Screen -1**

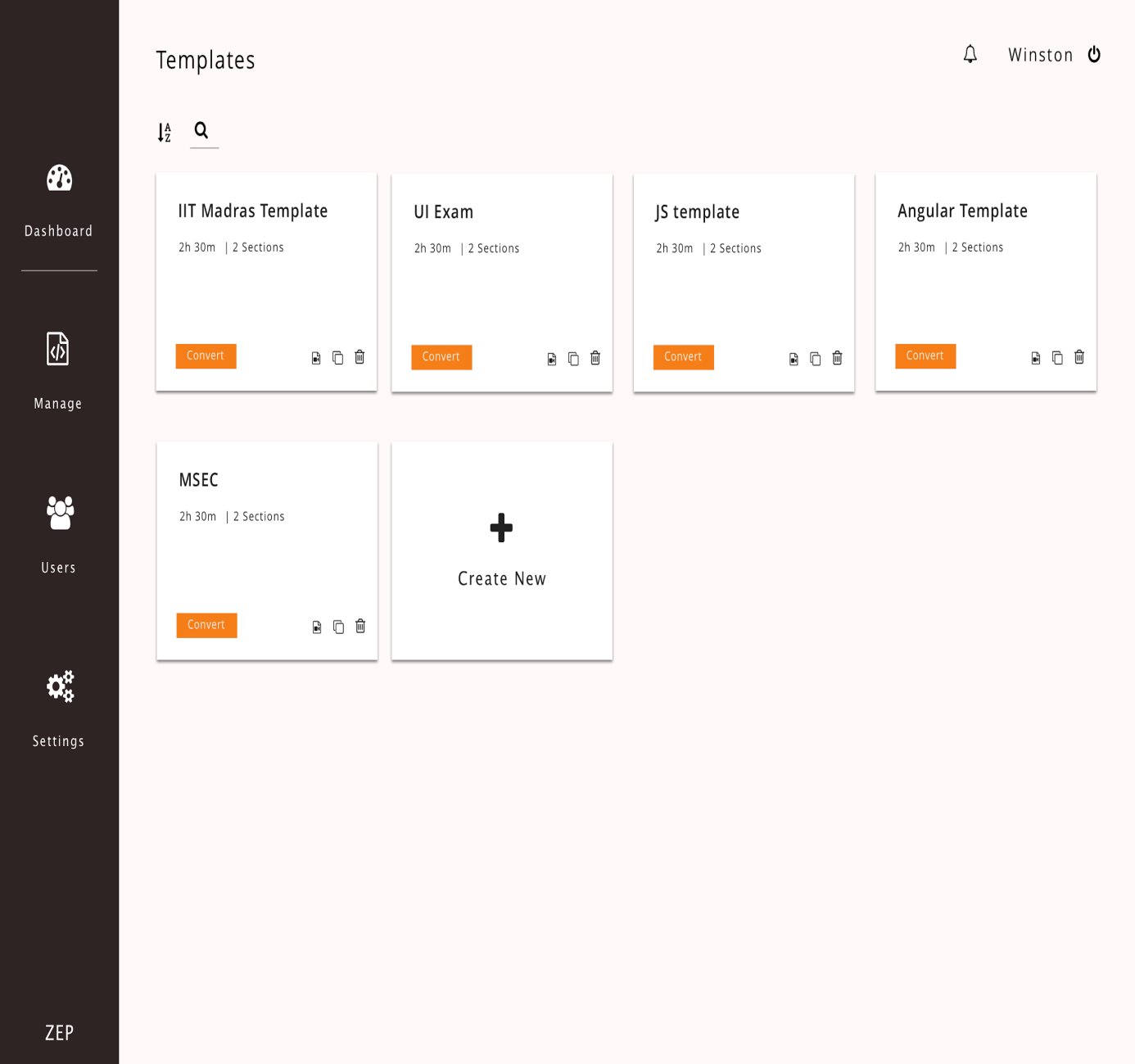
****

**Figure 6.4 Exam Screen-1**

**6.4.2 Exam Screen -2**



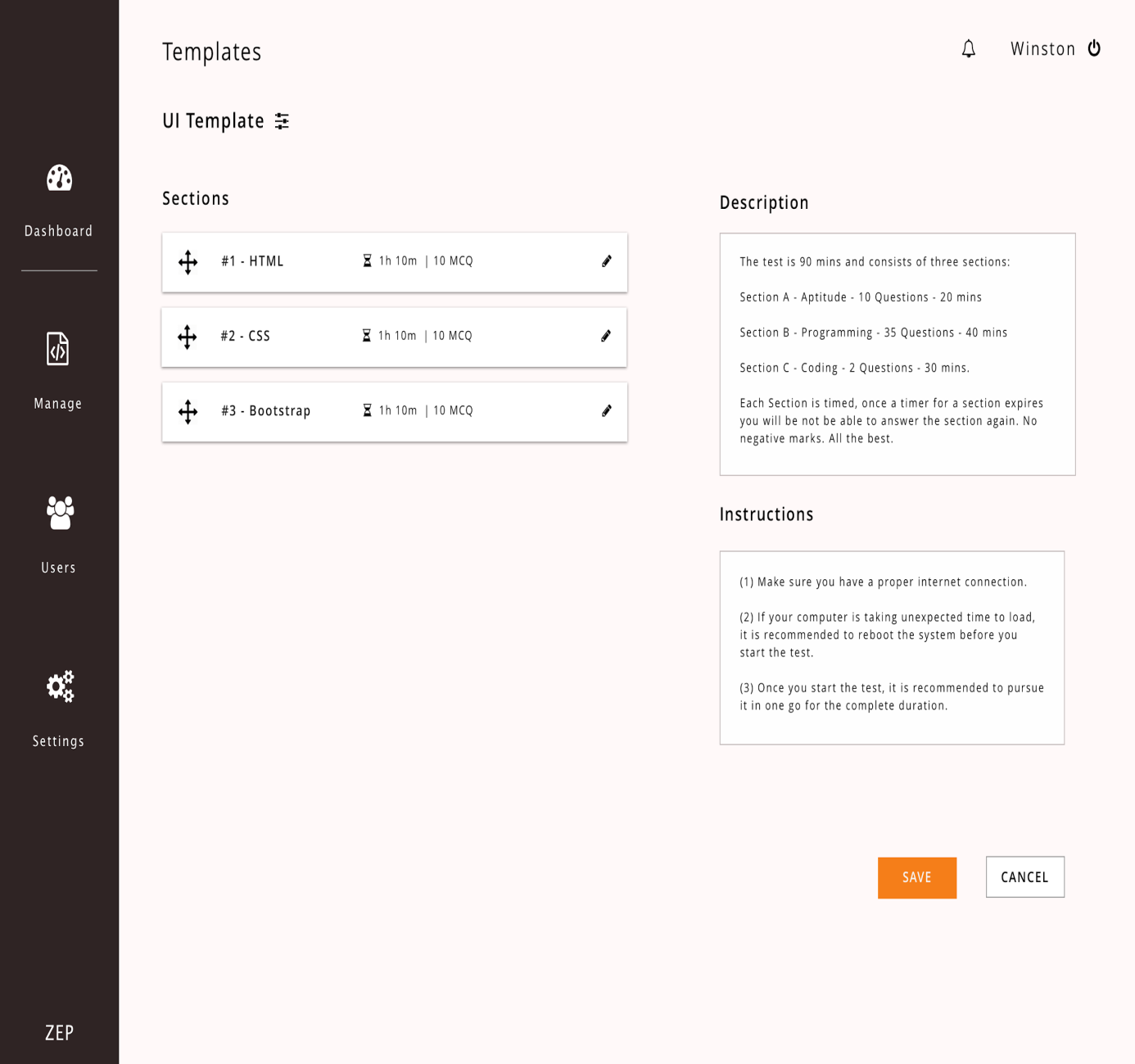
**Figure 6.5 Exam Screen-2**

**6.5 EXAM TEMPLATES LIST PAGE** 

**Figure 6.6 Exam Templates List Page**

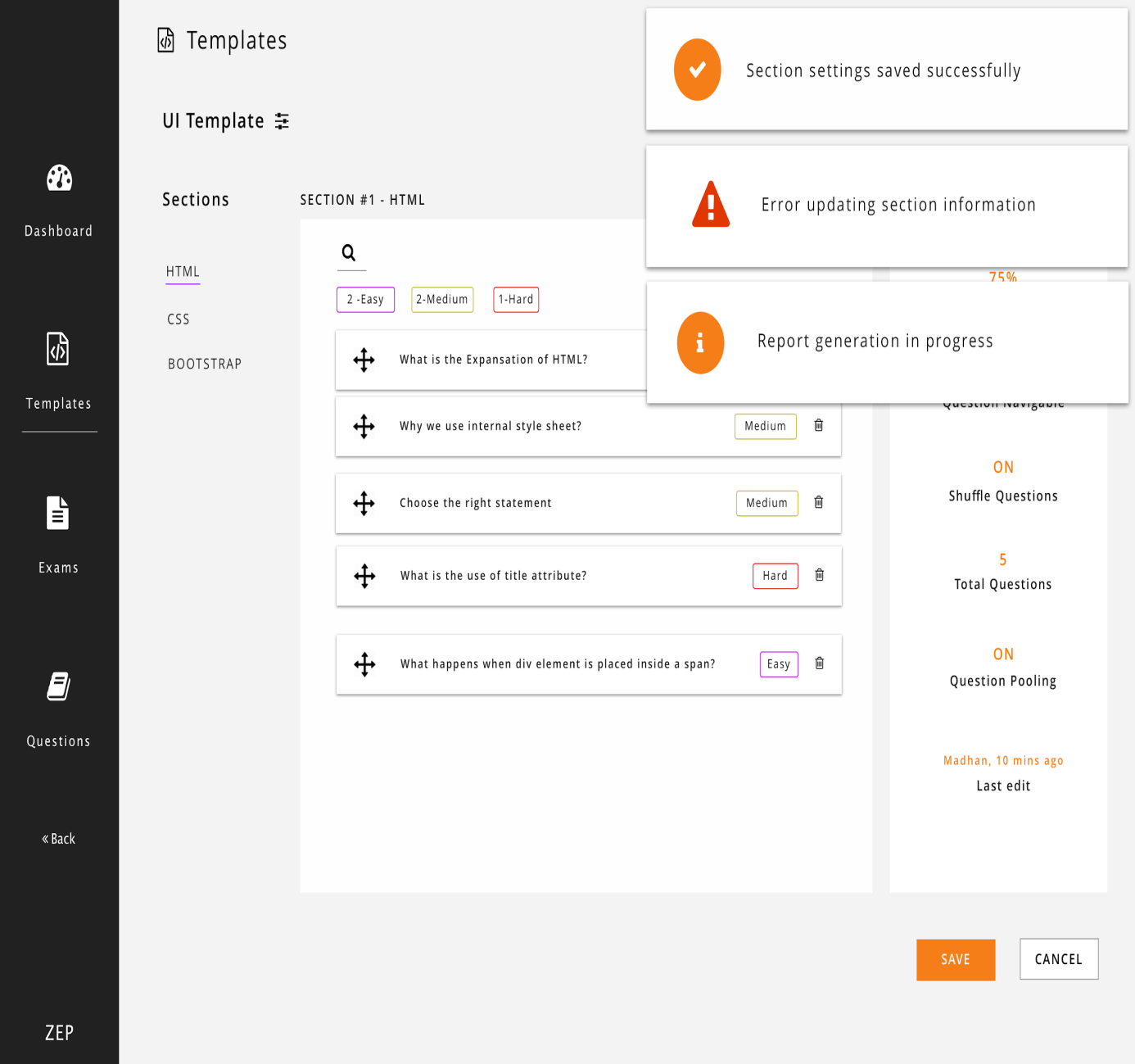
**6.6 EXAM TEMPLATE PAGE**

**6.6.1 Exam Template Page-1**



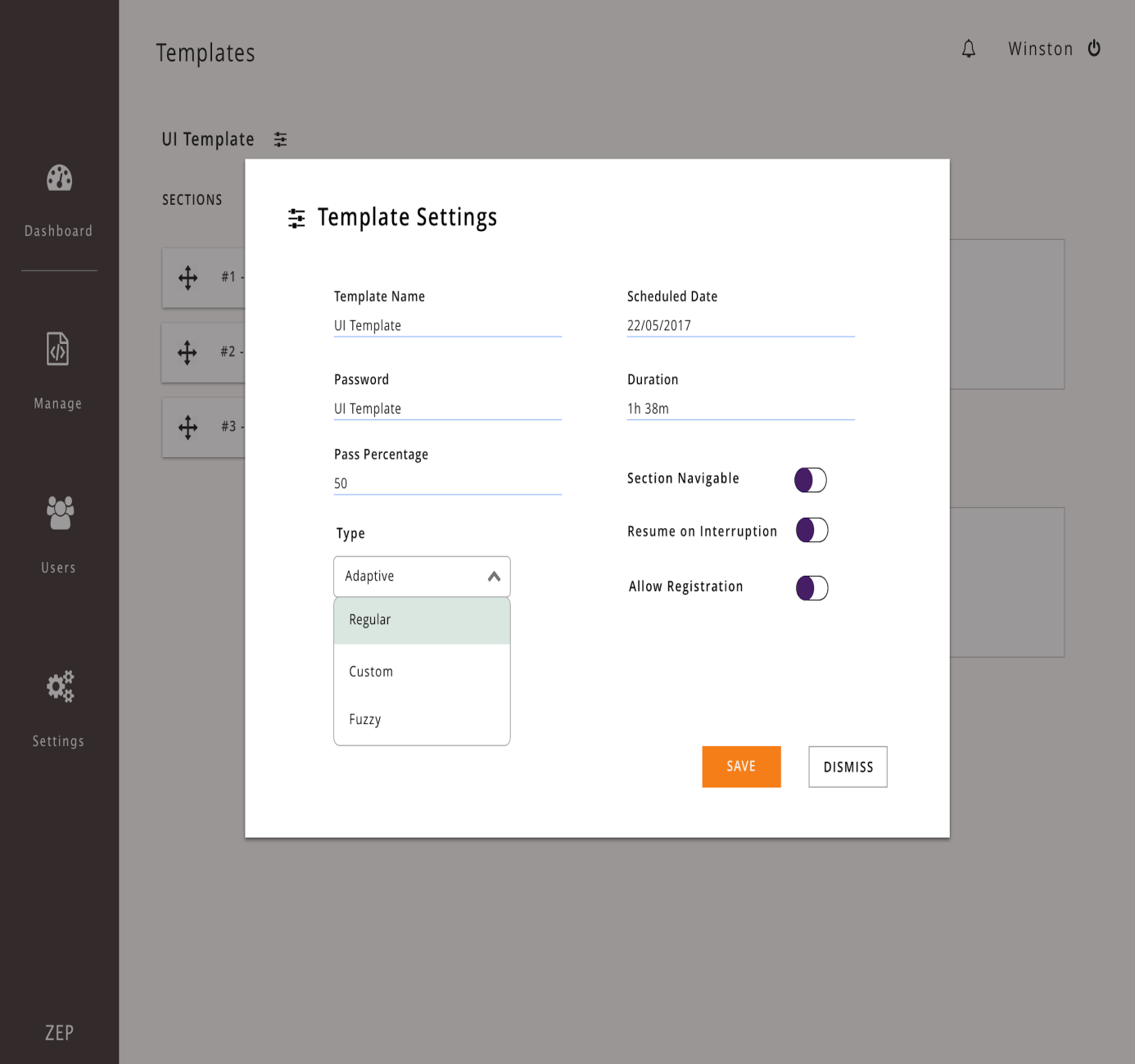
**Figure 6.7 Exam Template Page-1**

**6.6.2 Exam Template Page-2**

****

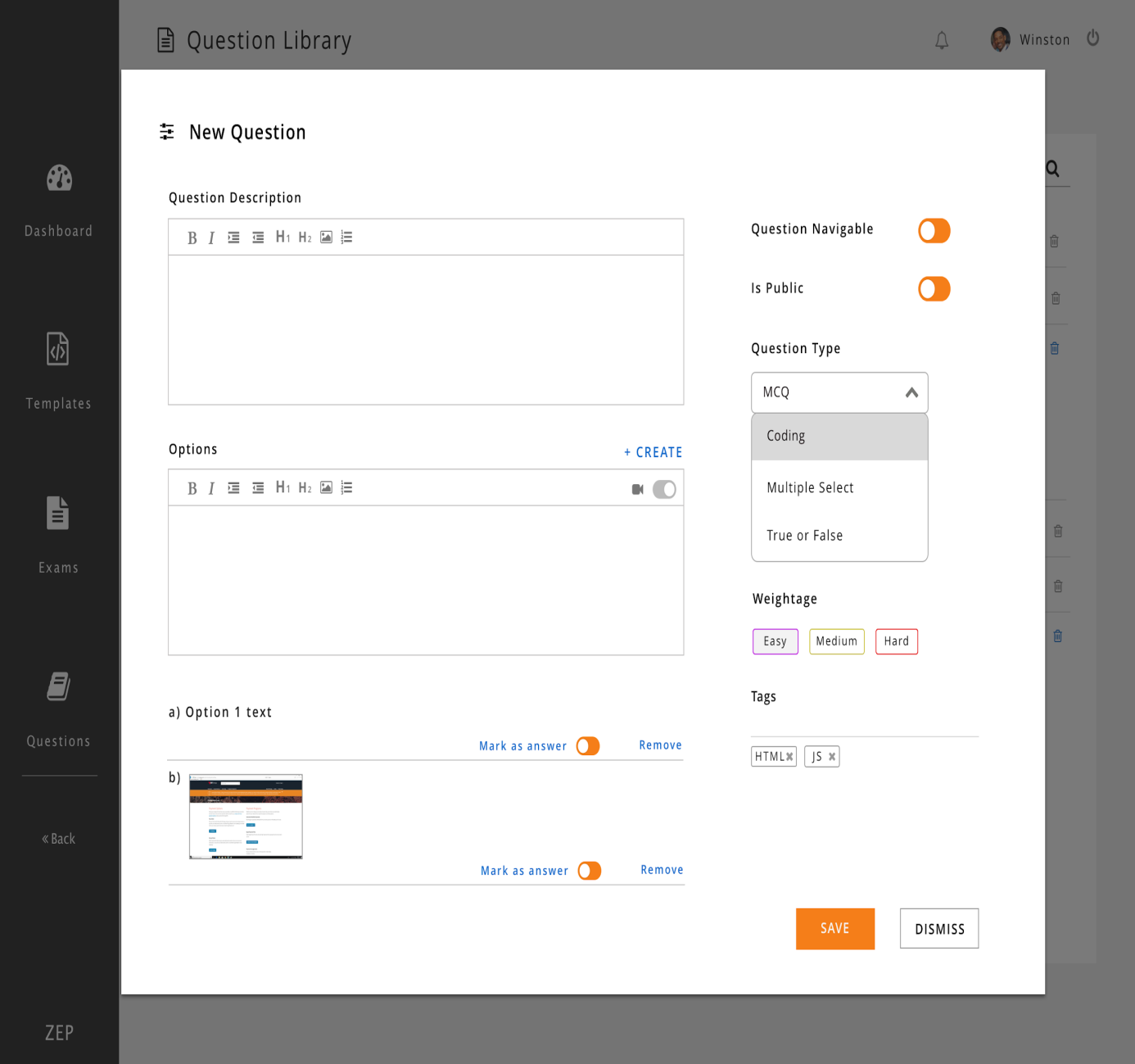
**Figure 6.8 Exam Template Page-2**

**6.7 EXAM TEMPLATE SETTINGS PAGE**



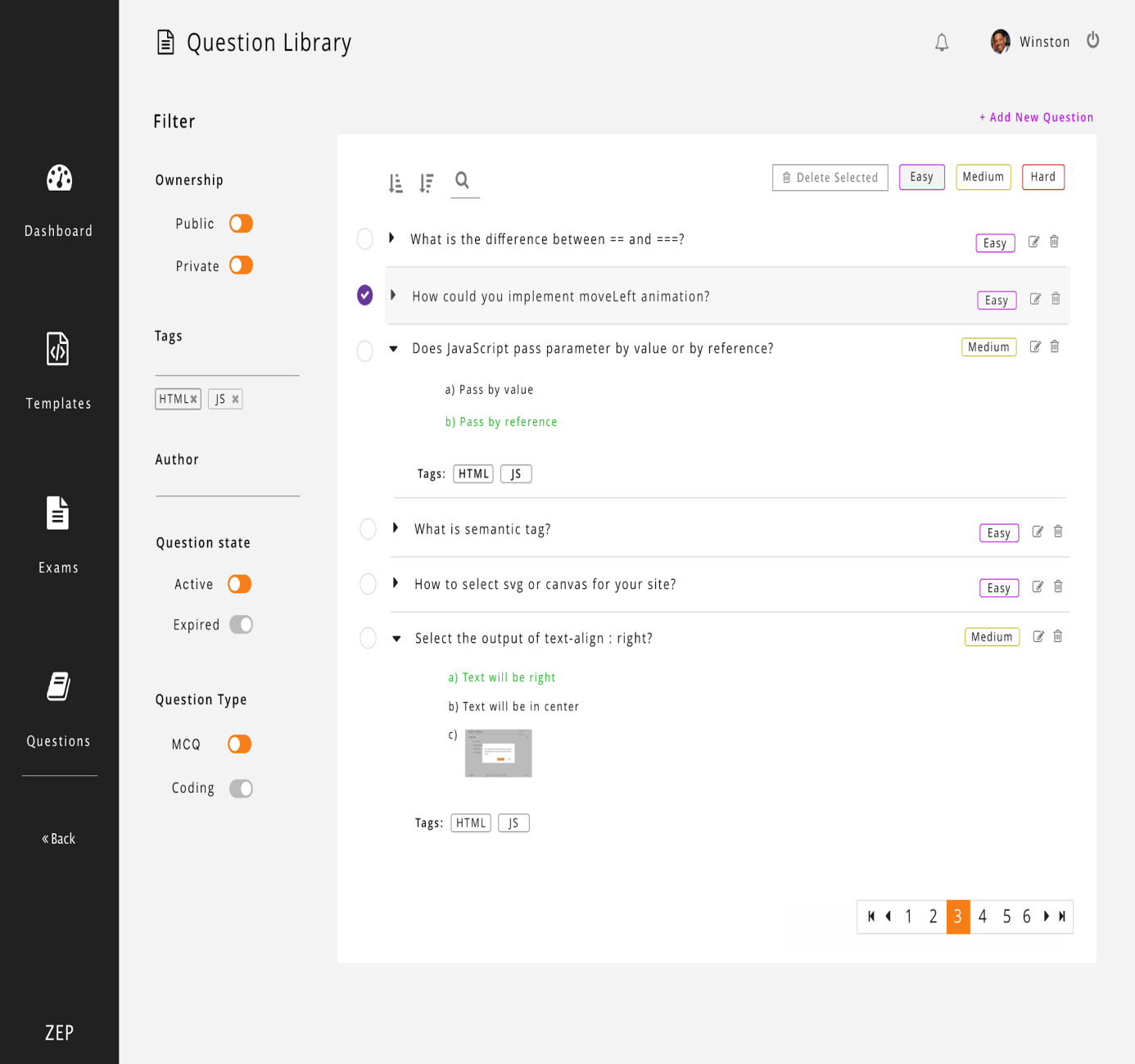
**Figure 6.9 Exam Template Settings Page**

**6.8 ADD QUESTIONS PAGE**

****

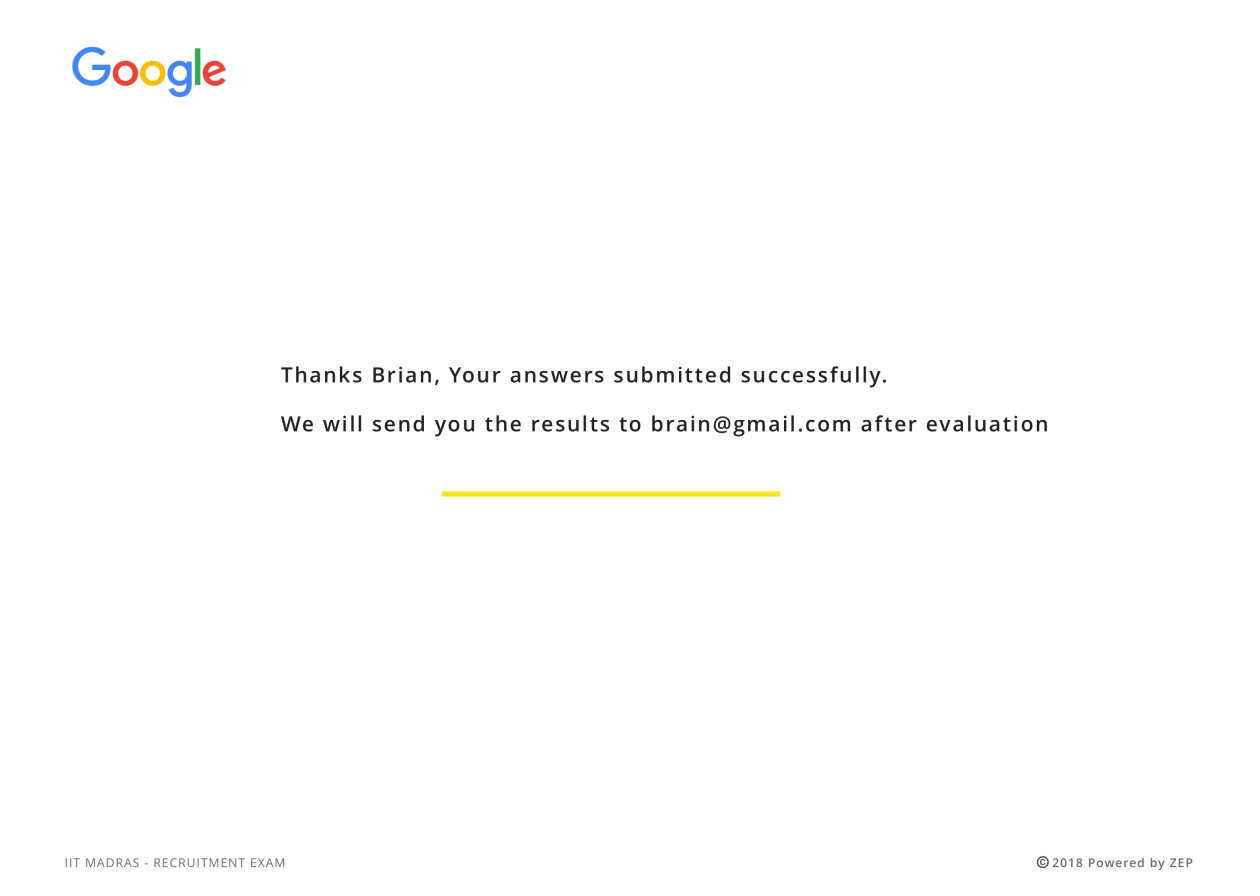
**Figure 6.10 Add Questions Page**

**6.9 QUESTION LIBRARY PAGE**

****

**Figure 6.11 Question library page**

**6.10 EXAM SUBMISSION PAGE**

****

**Figure 6.12 Exam submission page**

**CHAPTER 6**

**CONCLUSION AND FUTURE WORK**

The adaptive assessment is developed to evaluate student performance and the developed system overcomes the drawbacks of the existing system which is not fully adaptive. Security measures are implemented to ensure the correct examinee attends the exam and the questions are given based on the answers of the student. The crashing of system is taken into account and session resumption is provided when needed. The examinee is prevented from navigating to other windows during the exam or changing the time duration. The assessment uses fuzzy logic to frame questions adaptively and hence filters students effectively.

The future scope is as follows:

* The work accomplished is that a single session is maintained for the user logging in with the same username and password. In typical web applications, user with a registered account can do two different logins from two different work locations and can maintain two concurrent sessions opened but, in some applications we want to limit the users to one session per account.
* Deliver exams securely for students with identity verification and cheating prevention algorithms. So that the verified candidates can take the exam from anywhere.

**REFERENCES**

1. Amin Zollanvari, Refik Caglar Kizilirmak, Yau Hee Kho, Daniel

Hernández-Torrano (2017) ‘Predicting Students’ Gpa And Developing

Intervention Strategies Based On Self-regulatory Learning Behaviors’,

IEEE Access Vol.5, pp. 23792-23802.

2. Anastasios Karakostas, Ioannis Magnisalis, Stavros Demetriadis.

(2011) ‘Adaptive And Intelligent Systems For Collaborative Learning

Support: A Review Of The Field’, IEEE Transactions on Learning

Technologies Vol.4, pp. 5 - 20.

3. Antonio Garrido, Lluvia Morales (2014) ‘E-learning And Intelligent

Planning: Improving Content Personalization’, IEEE Revista

Iberoamericana de Tecnologias del Aprendizaje Vol.9, pp: 1-7.

4. JianPing Zhang, Bo Mu, Shipin Chen (2009) ‘Design and Application

of a Response-Driven Adaptive Test System’,IEEE Information

Technology and Computer Science International Conference.

5. Jie Xu, Kyeong Ho Moon and Mihaela van der Schaar (2017) ‘A

Machine Learning Approach for Tracking and Predicting Student

Performance in Degree Programs’, IEEE Journal Of Selected Topics In

Signal Processing Vol.11, pp. 742-753.

6. G. Pankaj Jain, Varadraj P. Gurupur, Jennifer L. Schroeder, Eileen D.

Faulkenberry (2014) ‘Artificial Intelligence-based Student Learning

Evaluation: A Concept Map-based Approach For Analyzing A Student's Understanding Of A Topic’,IEEE Transactions On Learning

Technologies Vol.7, pp. 267-279.

7. https://www.docker.com/what-docker

8. http://microservices.io/

9. https://jwt.io

10. http://couchdb.apache.org/

11. https://redis.io/