




# SOFTWARE AUDIT REPORT

## FOR

# SRI BALAJI VIDYAPEETH

DRAFT VERSION - 1.0

SUBMITTED ON - 12/4/2018

  
REGISTRAR  
SRI BALAJI VIDYAPEETH  
(Deemed University u/s 3 of UGC ACT, 1956)  
Accredited by NAAC with 'A' Grade  
NH 45-A, Pillaiyarkuppam, Pondicherry-607 402.

---

## CONTENTS

1. INTRODUCTION
2. OBJECTIVE
3. METHODOLOGY
4. INFERENCES
5. CONCLUSION

REGISTRAR  
SRI BALAJI VIDYAPEETH  
(Deemed University u/s 3 of UGC ACT, 1956)  
Accredited by NAAC with 'A' Grade  
NH 45-A, Pillaikuppam, Pondicherry-607 402.

## 1. INTRODUCTION:

Sri Balaji Vidyapeeth is a Leading Medical Deemed University, accredited by National Assessment and Accreditation Council (NAAC) with 'A' Grade. To maintain the institutional information, manage the workflow and to generate the documents and reports SBV has its own in-house developed software (on-going) and customized open-source software. SBV called for software experts to do an audit on their in-house software named Garuda and Chakra which is to manage their student and faculty information. Twilight IT Solutions Private Limited has been selected for auditing their system and a team of two software engineers made a whole day field visit studying the system and methodology.

This documentation provides an overall evaluation of the quality & standards of the Hardware and Software in Sri Balaji Vidyapeeth with respect to the evaluation done by the software team from Twilight IT Solutions Private Limited.

## 2. OBJECTIVE:

The core objective of this software review is to decide the below points:

1. To decide whether the architecture of the software is sufficient to meet with the institution's present and future need.
2. To decide if any change is required in the development environment (Hardware and Software)
3. To decide if any change is required in the production environment (Hardware and Software)
4. To cover the missing security aspects of the software.

### 3. METHODOLOGY:

A team of two software engineers proficient in software development, security and networking of large applications made a field visit to the development center interviewing the team. The IT infrastructure and IT Team was also interviewed to have a better understanding on the production environment, backup and recovery policy, security aspects, network traffic and campus connectivity.

The source code was analyzed to study the security aspect and standardization used in development. The development environment and the deployment methodology was studied. The team was interviewed on the architecture, development methodology, security measures, load balancing and other aspects.

### 4. INFERENCES:

#### ➤ ARCHITECTURE

The application is designed as a 3-tier application where the presentation layer and business layer has been segregated. The REST api with json data-interchange has been designed as per the industry standard and is secured with tokens. The best side of the architecture is that data exchange can happen with other applications which can consume api calls. It also seems that SBV is hosting other open source applications like Learning Management System, e-portfolio etc., where data flow can easily happen between these applications in future.

The chosen architecture is best suited for any University where multiple applications will be used for different purpose and data flow will be easy and secure. By adopting multi-tier architecture this application leverages additional benefits namely Increased security by introduction of a middle layer and extra level of indirection between front-end and database, safe from hacking, increased performance, scalable and enhancement.



If the University is looking for extending the application, they can use the existing api, hence reuse of code is well addressed. Also the api supports both token based and key based authentication and the same are issued for certain sections, therefore ensuring high security. This unique feature of the application addresses security and ensures safety of the data in all aspects. Further as per industry recommendation database is also isolated and no direct access to database is provided.

**RECOMMENDATIONS & SUGGESTIONS** : No Recommendations or Suggestions on the chosen architecture

➤ **TECHNOLOGY USED**

**MEAN Stack** is the technology used for the application. It is a good choice because SBV has an open-source policy and MEAN is the best open-source full stack in the market with large community support. Now a days most of our clients turn towards open-source in order to reduce licensing and other costs. MEAN stack is the most preferred technology by many enterprise leaders.

Normally MEAN Stack is chosen for its rich eco-system, excellent performance, increased flexibility and efficiency and of course for the reduced development cost (No developer license cost is associated except for any specific tools or libraries)

● **FRONT END**

Angular JS V 1.0, Material Design, AngularJS = **MVC** is used in the application. Template is used for this layer. Flex layout has been adopted for device compatibility. The application is more intuitive as it makes use of HTML as the declarative language and it is less brittle for re-organizing. Angularjs is the best comprehensive solution for rapid front end development since it has a wide range of features that include Restful actions, data building, dependency injection, enterprise-level testing and many more. Data models in Angular are

plain old JavaScript objects (POJO) and do not require extraneous getter and setter functions and makes the code to look clean and robust. Selection of **Angularjs** is a viable move.

- **TEMPLATE**

Triangular Template - A unique material design admin template which is built with **Angularjs**. From the ground up, this template has been built with Google's own Angular design material project.

Language support - This Template is multi language compatible with fully translatable specification. Translation is done via Angular translate & yandex translation service.

Device Compatibility - This template is Responsive and support almost all universal devices effectively.

Browser Compatibility - Triangular Template supports IE10, IE11, Firefox, Safari, Opera & Chrome.

Coding Standards - Clean code that compiles with John Papa's best practice Angular recommendations.

### **Recommendations & Suggestions :**

1. Twilight recommends to upgrade the front end to Angular 5 in future to stay in pace with the industry

### **> BUSINESS LAYER**

Inferred that Node js has been used as server scripting language. Business Layers are separated - followed best practice by separating auth and business logic. Dependency injections are used and followed the best industry standards. Coding standards are clean following the eslint tool.

- Security

Security is well addressed in this layer. All the request are authenticated with valid token and even api key is also authenticated with tokens.

## ➤ DATABASE

The database selection is MongoDB (Mongoose) ODM - V 3.2. Choosing of MongoDB which is one of the **NOSQL** is a great idea. As the details are maintained in a text files from the beginning and querying is made better and simple here. MongoDB offers out of the box, scale out and data localization with automatic sharding & easy maintenance of replica sets. The datasets has been neatly maintained data duplication has been effectively maintained by generating guide's.

**Recommendations and Suggestions:** Twilight suggests to upgrade database to V 3.6 to leverage its full capacity.

## ➤ SECURITY

Handling data security is one of the most important aspect of any application . In this application security is well addressed except few minor points.

Authentication method used : Token based authentication is used for all the request to the middle layer. For other applications, an api key is being generated by the administrator with certain access rights preventing them to access or modify data unless otherwise needed.

Token Blacklisting or Whitelisting : To avoid token hacking either blacklisting or whitelisting of tokens has to be implemented. This application implements token whitelisting using Redis server which is very lightweight in-memory data store thereby blocking inactive tokens. Tokens are made to expire automatically after



24 hrs, so users has to login again to regain fresh tokens. Refresh tokens are available only for mobile users which is also addressed well in the application.

Encryption of Token : The token generated is encrypted and handled by JWT.

User Authentication : All users are authenticated by a username and password. Resetting password is done by getting OTP to their personal mobile or email Id. User accounts are kept secured.

Password Encryption : Password is encrypted and stored in the database.

Brute force attack handling : As of now brute force attack is not addressed by the application

Handling of Sensitive Data : Certain data like exam marks are kept secured and locked and editing is allowed only through authorization by authorities through OTP to their mobile.

Injection : Query Injection has been addressed well in the code

Database : Database is well secured by a password and kept isolated and direct access to database is not allowed thereby securing the database. The database server is not connected to internet and access is only through the api which is as per the industry recommendation.

Firewall and Antivirus software : The servers are well protected by firewall and anti-virus software. cyberoam is being used as firewall and in future SBV has planned for migrating to sophos. Currently Bitdefender and endpoint are used as antivirus software



User Access Control : This application uses both Claim Based Access Control (CBAC) and Role Based Access Control (RBAC). RBAC is used for routing to the dashboard and CBAC is used for module accessing.

Secured Socket Layer : Not implemented

### Recommendations & Suggestions :

1. Brute force attack should be implemented by locking users after certain unsuccessful attempts.
2. SSL Certificate should be implemented

#### ➤ LOGS

Logs are very important for any application. This application maintains every day text logs and no time limit is set and since this is an ongoing project the team prefer to manual clearing of logs for continuous monitoring. Important activity are logged separately using bunyan library. The current log system is found efficient.

#### ➤ LOAD BALANCING

Load balancing has not been implemented. Since current concurrent users are less than 100 load balancing is not required.

When concurrent users reach 500 **we strongly recommend to implement load balancing.**

#### ➤ MAJOR LIBRARIES USED

**JWT** - JSON Web Token used for token based authentication

**Async** - Asynchronous API handling

**Body-parser** - Used for parsing user request

- Excel js** - Processing excel file
- Ftp-client** - FTP file handling
- Googleapis** - Integrating google APIs
- Html-pdf** - Convert html template to pdf document
- Multiparty** - Upload files to server
- Node-cron** - Running cron jobs
- Node-zip** - Generating compressed zip file
- Nodemailer** - Generate and process emails
- Q** - Handling asynchronous API as synchronous manner
- Uuid** - Generate Universal unique identifier
- Underscore** - Perform utility tasks
- Request** - Send Web API request and handling response

The libraries leveraged are of industry standard and recommended for the Technology.

#### ➤ **DEVELOPMENT METHODOLOGY**

**Agile Methodology** is implemented at MGMC which is a very futuristic concept in SDLC. Agile Software Development is an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto. Solutions evolve through collaboration between self-organizing, cross-functional teams utilizing the appropriate practices for their context.

- **SOURCE VERSIONING AND REPOSITORY** - At SBV GITLab private repository is leveraged and the codes are safe and secure. As a best practice master should not be touched and coders has to to work in branches. Creating and switching to branches will help to have a development code and bug free "Master".
- **CODE EDITOR** - Visual Studio Code and Atom.io is a perfect choice.
- **PROJECT MANAGEMENT TOOL** - The observed project management tools leveraged includes Google to-do task, email & agenda. We strongly

recommend to utilize the exclusive tools which is more specific for Project management to maintain agility.

- **DOCUMENTATION** - Currently there is no documentation process followed. It is better and recommended to do minimal documentation that adds value to the existing system.
- **RELEASE MANAGEMENT** - Currently there is no release management tool leveraged. It is recommended to stick with this in future to manage Software releases better.
- **TESTING - Jasmine**, an open source testing framework for JavaScript is leveraged for the Test cycle. This is not sufficient and Manual testers are indeed to streamline the testing process. **We strongly recommend that dedicated testing team should be involved to run through the application before deployment**
- **DEPLOYMENT PROCEDURES** - Development is done in local and build is directly acquired from the repository in production. Deployment is done as per the checklist they maintain. Based on the frequency of the deployment, this can still be optimized.
- **STAGING ENVIRONMENT** - We inferred that there is No Staging Environment implementation.

➤ **PRODUCTION ENVIRONMENT**

○ SERVER

Brief details on Server:

**Front End Server (VM)**

Dedicated CPU Cores	-	6
Dedicated RAM	-	6gb
Network Port	-	1 g
HDD	-	500gb

**Database Server (VM)**

Dedicated CPU Cores	-	4
Dedicated RAM	-	6gb
Network Port	-	1 g
HDD	-	500gb

**Backup & Recovery:**

1. Daily Auto Backup of entire VMs to ( Network Storage )
2. Once in two days backups are copied to "Tape Library" ( Backup set 1)
3. Monthly backups are copied to "Tape Library" ( Backup set 2)

Front-end and middle-tier is hosted in a single server and database in a separate server. In case of high load needs, we suggest to separate Front-end / middle-tier and host it in separate servers.

- o **NETWORK** - Fully secured intranet highly accessible to the SBV staffs
- o **BANDWIDTH** - Currently the observed bandwidth is 10 mbps. We strongly recommend that at least 50 mbps dedicated line has to be provided to the server for efficient orientations or it is also better to move to cloud.

➤ **DEVELOPMENT MACHINES UPGRADATIONS** - Our team drilled more deep into the systems/software that has been used for the development. For hassle free development, Twilight recommends to upgrade the machines / systems to **8GB RAM with i5/i7 processor.**

➤ **3rd PARTY TOOLS USED**

Our team understood that either pirated or trial / free version of 3rd party tools are used in the application. We strongly recommend to go for a paid and licensed version.

REGISTRAR  
**SRI BALAJI VIDYAPEETH**  
 (Deemed University u/s 3 of UGC ACT, 1956)  
 Accredited by NAAC with 'A' Grade  
 NH 45-A, Pillaiyarkuppam, Pondicherry-607 402.



5. CONCLUSION:

According to the software / review done by our team, we feel that the Software Development process, architecture & quality seems to be on the industry standards.

The technological stack selection, Agile development methodology implementations and robust data recovery system make the entire process streamlined and be more futuristic.

We trust our suggestions and recommendations helps you to strengthen your system better and be more scalable.

  
REGISTRAR  
SRI BALAJI VIDYAPEETH  
(Deemed University u/s 3 of UGC ACT, 1956)  
Accredited by NAAC with 'A' Grade  
NH 45-A, Pillaiyarkuppam, Pondicherry-607 402.