

**Web App for Faculty Approval Management**

**A PROJECT REPORT**

***Submitted by***

**GANESH SETHU(18BCS037)**

**SUBASH C (18BCS061)**

**SARAN N (18BCS098)**

***In partial fulfillment for the award of the degree***

*of*

**BACHELOR OF ENGINEERING**

in

**COMPUTER SCIENCE AND ENGINEERING**

**KUMARAGURU COLLEGE OF TECHNOLOGY**

**COIMBATORE-641 049**

(An Autonomous Institution Affiliated to Anna University, Chennai)

**January 2022**

**KUMARAGURU COLLEGE OF TECHNOLOGY**

**COIMBATORE 641 049**

(An Autonomous Institution Affiliated to Anna University, Chennai)

**BONAFIDE CERTIFICATE**

Certified that this project report **“Web App for Faculty Approval Management”** is the bonafide work of “**GANESH SETHU (18BCS037), C.SUBASH (18BCS061) and N.SARAN(18BCS098)”** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **Dr. Devaki. P, Ph.D.,** |  |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Department of Computer Science and  Engineering, | Department of Computer Science and Engineering, |
| Kumaraguru College of Technology | Kumaraguru College of Technology |
| Coimbatore – 641 049. | Coimbatore – 641 049. |

The candidates with University register number **18BCS037, 18BCS061, 18BCS098** were examined in the Project Viva-Voce examination held on …………………

Internal Examiner External Examiner

**DECLARATION**

We affirm that the project work titled **“WEB APP for FACULTY APPROVAL MANAGEMENT”** being submitted in partial fulfillment for the award of B.E Computer Science and Engineering is the original work carried out by us. It has not formed the part of any other project work submitted for the award of any degree or diploma, either in this or any other University.

**GANESH SETHU (18BCS037)**

**SUBASH C. (18BCS061)**

**SARAN.N(18BCS098)**

I certify that the declaration made above by the candidates is true.

----------------------

Assistant Professor,

Department of Computer Science and Engineering,

Kumaraguru College of Technology,

Coimbatore – 641 049.

**ACKNOWLEDGEMENT**

We express our profound gratitude to the management of Kumaraguru College of Technology for providing us with the required infrastructure that enabled us to successfully complete the project.

We extend our gratitude to our Principal, **Dr. D. Saravanan,** for providing us the necessary facilities to pursue the project.

We would like to acknowledge **Dr. P. Devaki,** Professor and Head, Department of Computer Science and Engineering, for her support and encouragement throughout this project.

We thank our project coordinator **Dr. L. Latha,** Professor, Department of Computer Science and Engineering and guide **Dr. Uma Maheswari,** AssistantProfessor, Department of Computer Science and Engineering, for their constant and continuous effort, guidance and valuable time.

Our sincere and hearty thanks to staff members of Department of Computer Science and Engineering of Kumaraguru College of Technology for their well wishes, timely help and support rendered to us during our project. We are greatly indebted to our family, relatives and friends, without whom life would have not been shaped to this level.

**GANESH SETHU**

**C.SUBASH**

**N. SARAN**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No** | **Title** | **Page No** |
|  | **ABSTRACT** | **2** |
| **1** | **INTRODUCTION** | **3** |
|  | 1.1 CONCEPTUAL STUDY OF THE PROJECT | 3 |
|  | 1.2 OBJECTIVES OF THE PROJECT | 3 |
|  | 1.3 SCOPE OF THE PROJECT | 4 |
| **2** | **LITERATURE REVIEW** | **4** |
|  | 2.1 LITERATURE REVIEW OF JOURNALS | **4** |
| **3** | **PROBLEM DEFINITION** | **6** |
|  | 3.1 DISADVANTAGES OF THE EXISTING SYSTEM | **6** |
| **4** | **PROPOSED SYSTEM** | **7** |
|  | 4.1 METHODOLOGY | **8** |
|  | 4.2 FLOW DIAGRAM | **9** |
|  | 4.3 IMPLEMENTATION | **10 - 20** |
| **5** | **SYSTEM REQUIREMENTS** | **21** |
|  | 5.1 USER SIDE REQUIREMENTS | **21** |
| **6** | **CONCLUSION** | **21** |
| **7** | **APPENDIX** | **2** |

**ABSTRACT**

This paper proposes an archival and approval system wherever hierarchy of staff is followed. So that juniors can be approved by senior members to attend events, conduct events and also after attending the events the system allows the users to upload the verification certificates and register it as an attended event. It allows management of the allocated budget for each person. The dashboard visualizes the data. The system uses MVC model and REST API to achieve the above-mentioned functionalities.

**1. INTRODUCTION**

**1.1 CONCEPTUAL STUDY OF THE PROJECT**

Archiving events and managing reports of the events attended by a staff is a common activity in every campus. But when it comes to large scales it is a very difficult task to handle it manually. The use of digital forms and storing and retrieving data in a spreadsheet is a tough process. In a world full of software products, it is inevitable to automate the existing manual process to a modern user-friendly solution to draw insights and conclusions from the generated data to understand the flow of the system and understand the events and nature of members to push them to their limits to attend more technical events and gain skills.

Our system uses REST API’s.It can be used over nearly any protocol, when used for web APIs it typically takes advantage of HTTP. REST API provides a great deal of flexibility. Data is not tied to resources or methods, so REST can handle multiple types of calls, return different data formats and even change structurally with the correct implementation of hypermedia. This flexibility allows developers to [build an API](https://www.mulesoft.com/lp/ebook/api/building-api-blueprint) that meets your needs while also meeting the needs of very diverse users.

**1.2 OBJECTIVES OF THE PROJECT**

To solve the inconveniences as mentioned in the existing system, an e-Archiving system (FAM APP - FACULTY ARCHIVAL and APPROVAL MANAGEMENT) is proposed. The proposed application contains the following features,

* Authentication
* Various Levels of Approval
* Automated Archiving
* Email Notification
* Dynamic report generation

**1.3 SCOPE OF THE PROJECT**

* Colleges where ever hierarchy of staff is followed
* Offices where seniors have to approve their juniors
* Event Management Organizations
* Companies where archival system is needed
* Schools where events, approval and archival is necessary.
* Automation of the pen-paper based approval systems in Institutions

**2. LITERATURE REVIEW**

[1] Gordon, David & Heaton, Jacob. (2008). Documentation accountability: Approval to archiving. 128-132. 10.1109/AUTEST.2008.4662598.

This journal paper proposes Information Technology (IT) has been used for a long time in support activities in different organizations. IT is used in the organization to make numerous contributions such as reducing time, cost, and supporting better services for customers. In the knowledge era, practitioners also considered IT to support KM. IT use in Knowledge Management (KM) in the various methods Knowledge management systems (KMSs) are a mature information technology often used to accumulate policies and procedures in an accessible framework. However, the usefulness of a KMS depends on its capabilities and the users: a KMS is only useful if it contains a certain quantity and quality of information. They highlight there are three important components in KM is about; first, is about managing people, it is related with networking; collaboration; the second is about managing knowledge/information, it is related to accessibility, searching, validating, taxonomy, up-to date, knowledge flooding and Managing Information technology is related with information/ knowledge security, speed, and reliability. It is believed that KMS give huge opportunities to break down barriers by making the information presented at every level and units in organization hence it will help to enhance organization and become more effective. They also add KM has been recognized by organizations for decades. On many KM implementations, most of the organizations focus their KM on knowledge creation and how it can be leveraged in their organization. A good KMS can increase information quantity by making documentation processes faster and can increase information quality by ensuring accountability throughout the testing process. This paper illustrates how the KMS, JARIS, can streamline each test phase, from approval to archiving, by reducing paper documentation and increasing information accountability and accessibility. In this study we would propose the framework that can be used for organizations to develop and implement KMS, especially focusing on supporting people interaction in virtual communities. Most KM researchers agree that knowledge does not exist in technical elements-they only exist in human beings who are able to act upon the knowledge. This paper concludes that KMS has a great opportunity to enable virtual communities in KM. The organization should pay attention both in social and technological aspects when developing and implementing KMS. As general KMS have two functions: managing people interaction and managing knowledge/information. The organization should carefully decide what type of KMS they should adopt. Each KMS has different treatments and factors for attention. The right decision of choosing tools for support and enabler KM practices in an organization is critical and would impact the benefit of the KM process.

[2] Sitnov, Vladimir & Senchenko, Alexander & Fatkin, George. (2020). Archiving System Software for LIA-20. Vestnik NSU. Series: Information Technologies. 18. 69-80. 10.25205/1818-7900-2020-18-3-69-80.

This journal proposes about the control system of LIA-20 which is based on the Tango control framework. HDB++ was used as it provided efficient storage using Postgres as a database. The EventSubscriber TANGO device server, or Archiver, is the archiving system engine.It will subscribe to archive events on request by the ConfigurationManager device. The EventSubscriber is designed to start archiving all the already configured Attributes, even if the ConfigurationManager is not running. Moreover, being a TANGO device, the EventSubscriber configuration can be managed with Jive.

[3] Sharma, Sushil & Sang, Sinawong & Lee, Jeong-Dong & LeeSeoul, Jongsu. (2011). The Case of Electronic Approval System. 10.4018/978-1-60960-597-1.ch014.

The main core of the journal suggests that factors that influence user adoption of e-Government services: the Electronic Approval System (EAS). This study uses the Technology Acceptance Model (TAM), the extended TAM (TAM2), the Diffusion of Innovation (DOI), and trust to build a parsimonious yet comprehensive model of factors that influence user acceptance of the EAS. We collected data from a total of 112 public officers in 12 ministries in Cambodia. We assessed the model with regression analyses. The findings in this article show that the determinants of the model (perceived usefulness, relative advantage, and trust) explain 30.5% of the variance in user acceptance of the EAS. At the same time, image, output quality, and perceived ease of use explain 38.4% of the variance in user perception of the usefulness of the EAS. In this article, we discuss our findings, implications, and suggestions for future research

**3. PROBLEM DEFINITION**

Currently the archival of events and generating reports is managed manually by-passing digital forms and storing the details in spreadsheets. No authentication and No data visualization are there in the existing system. So, User’s data is not secure and data can't be comprehended.

**3.1 Disadvantages of The Existing System:**

* It requires a lot of time to record or to retrieve the details.
* The person who has to record the details must perform their job very carefully. Even a small mistake could create a lot of problems.
* Security of information is very less. Report generation of all the information is a very tough task.
* Maintenance of individual reports is very difficult and all the operations must be performed in perfect manner without any degradation, else it may finally result in the failure of the entire system.

**4. PROPOSED SYSTEM**

To solve the inconveniences as mentioned in the existing system, an e-Archiving system (FAM APP - FACULTY ARCHIVAL and APPROVAL MANAGEMENT) is proposed. The proposed application contains the following features:

* **Security:**

Authentication of different users based on the hierarchy will be handled. Accessing contents will be restricted based on the user type.

* **Automated Archiving:**

The events attended by the faculty will automatically get archived and use of forms and spreadsheets will be completely removed.

* **Notifications:**

Email notifications to the end users will be pushed to track the approval and rejection of events.

* **Data Visualization:**

A Dashboard page will display the performance and participation of users and various graphical charts will be used to visualize the available data.

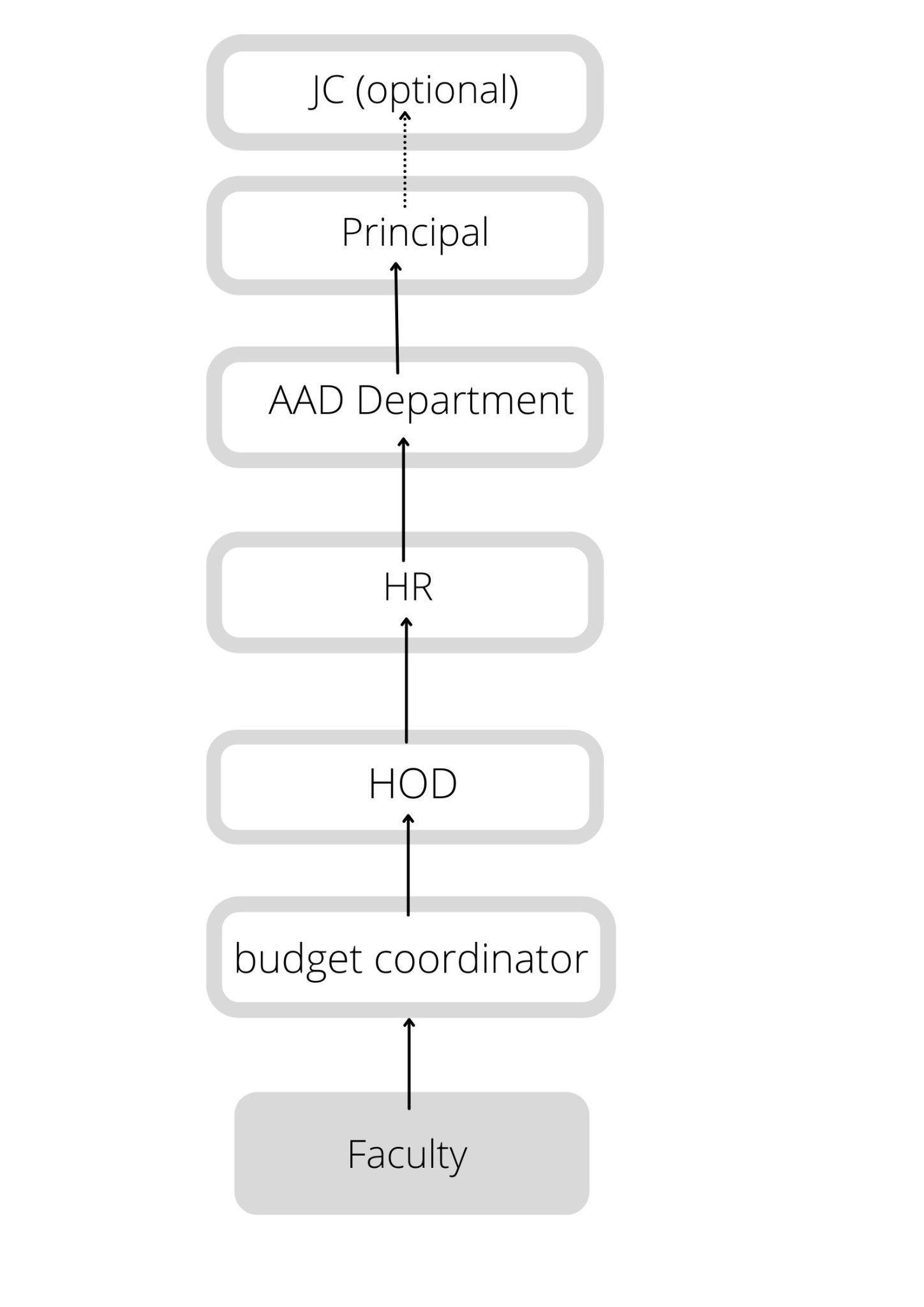
* **Backup and Recovery:**

A proper backup copy is stored in a separate system or medium, such as tape, from the primary data to protect against the possibility of data loss due to primary hardware or software failure.

**4.1. METHODOLOGY**

In this project we present a web-based application to solve the manual archiving and approval system maintained inside the campus. The system can authenticate different users and allow them to make event request and approve accordingly. Traditional Archiving systems used a digital form circulated within the campus and the data entries were maintained in a spreadsheet manually which is a tedious and difficult task as there is no central application where everything can be managed, maintained and archived at one place. Our Web based solution provides a solution to the above issues by automatically archiving the events and manage the approval process in a more efficient manner. Thus, our web application can be used in any institution where the hierarchy of staff members is followed. The senior staff members can approve or disapprove their juniors to attend or organize various events considering the budget, date/time, outcome of the event etc... Different types of reports can be generated based on department, users, dates or events.

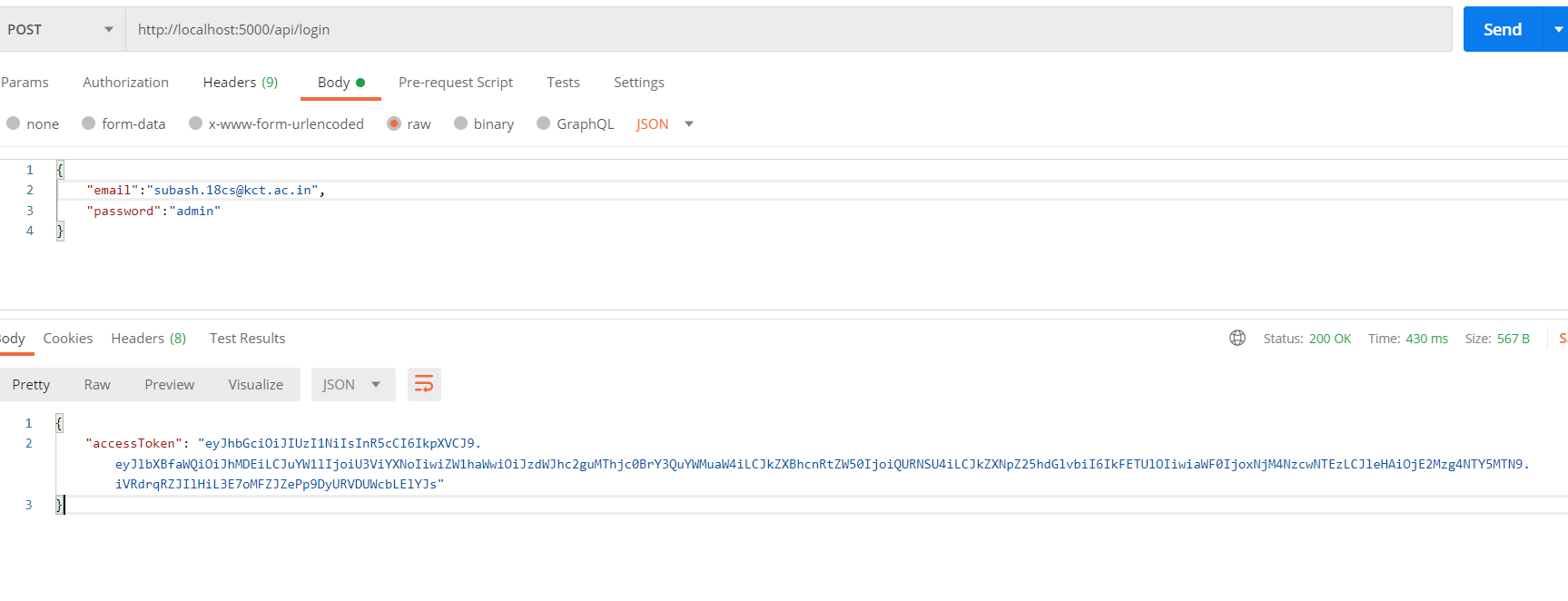
**4.2. FLOW DIAGRAM**

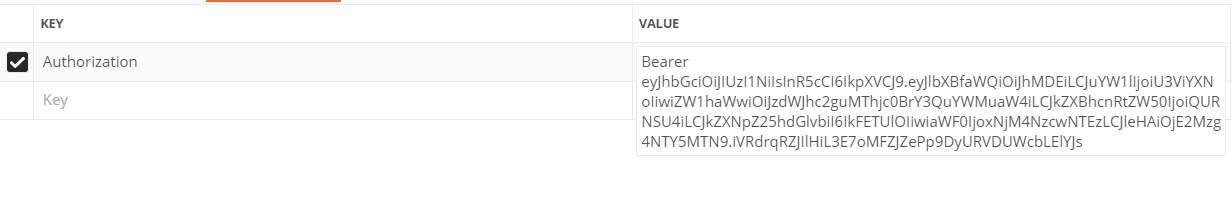


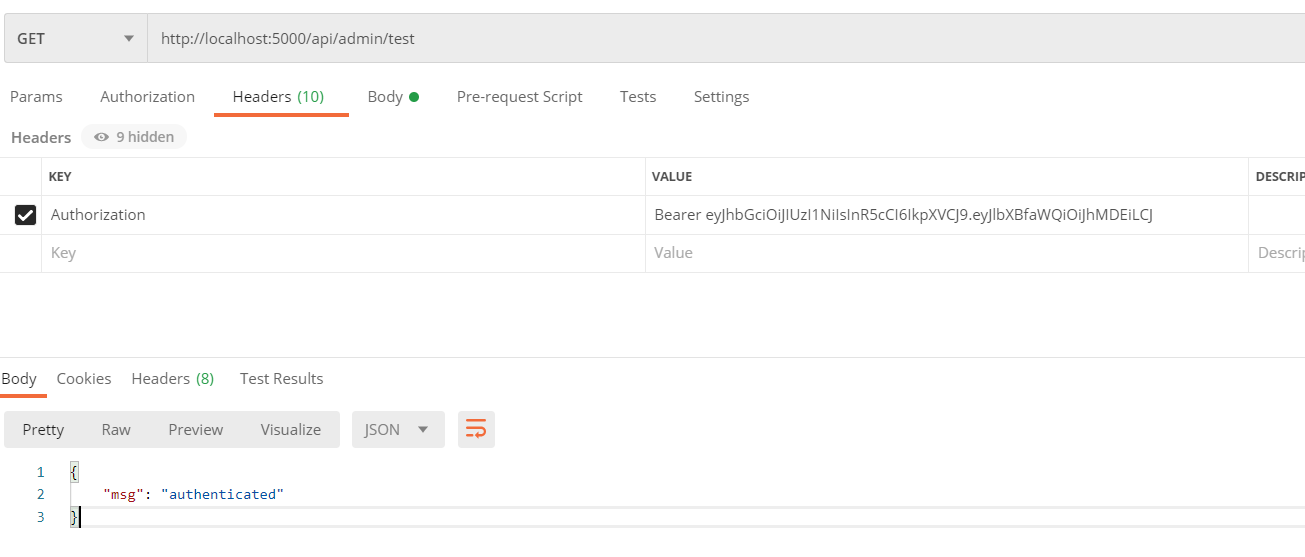
**4.3. IMPLEMENTATION**

The implementation started with developing the REST API’S with node js and express js which is a nodejs framework used for server-side applications. We use MySQL database to store and retrieve data. For authenticating different used we used JWT - JSON Web Token to authenticate and authorize different users. Testing of REST API is done using the POSTMAN tool.

**Authentication using JWT :**

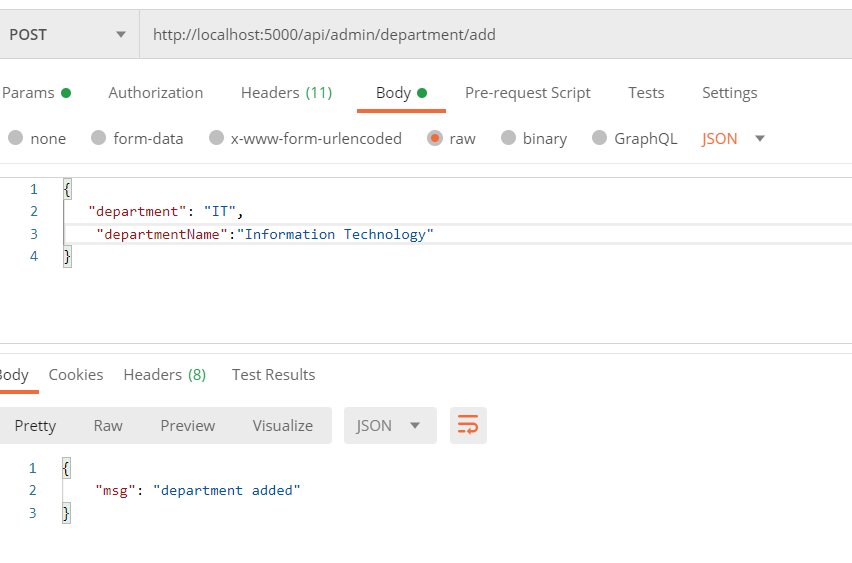
****

****



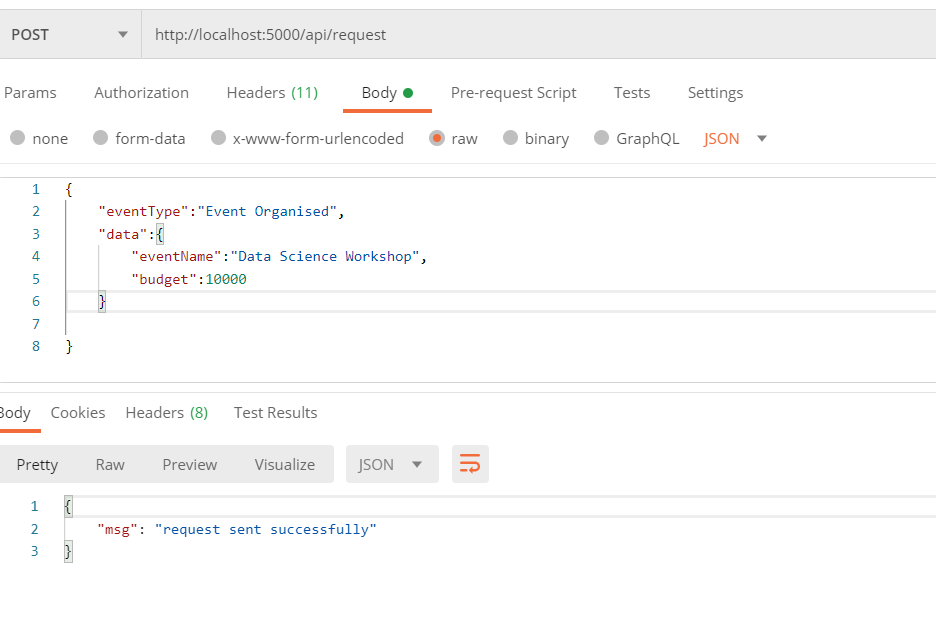
**Sample admin routes :**

There are some features which are used only by administration users like adding departments , adding new user , removing or updating users etc..

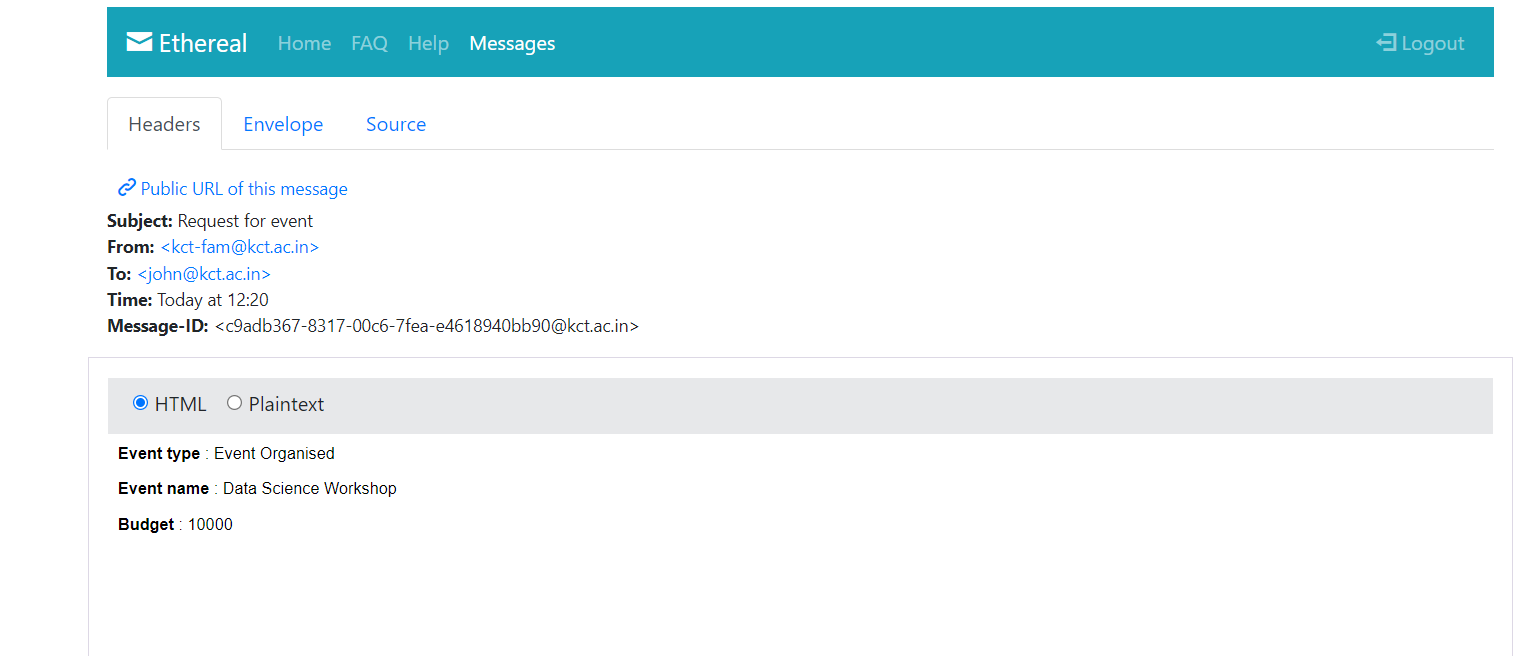


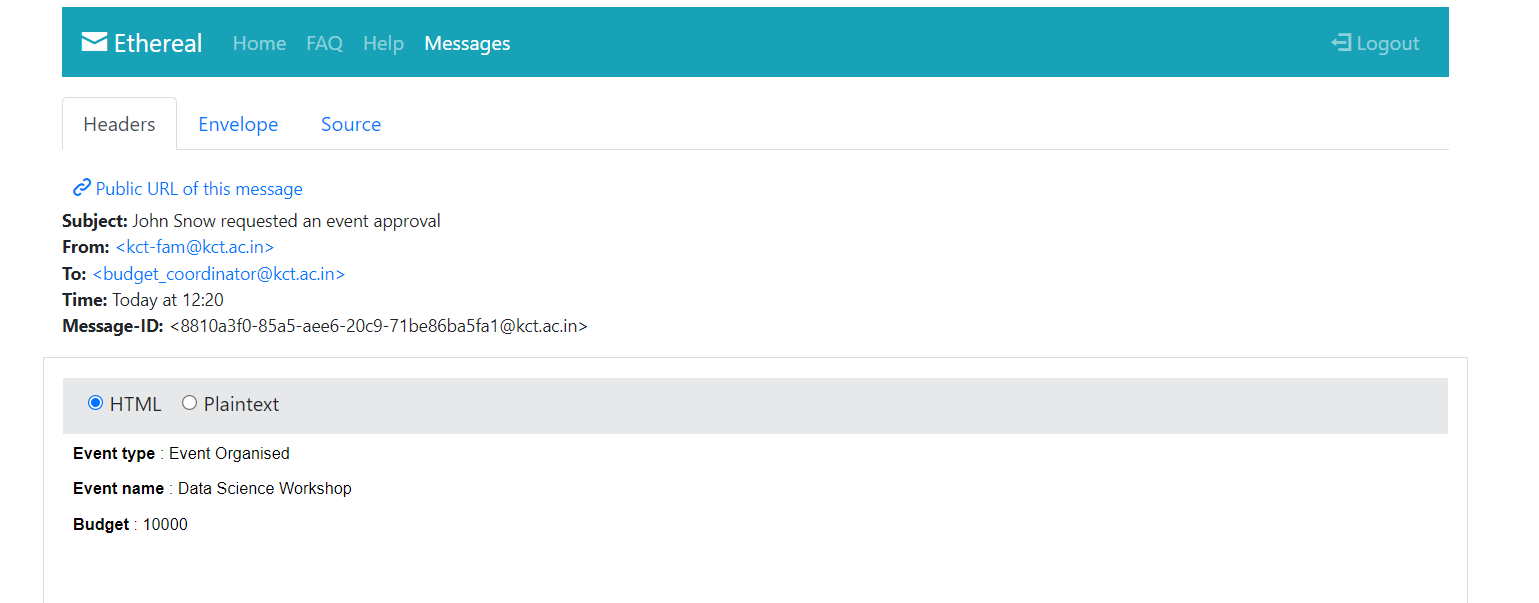
**Event request:**

Any new event requested must follow the hierarchy mentioned above and complete all levels of approvals before creating itself as an event.

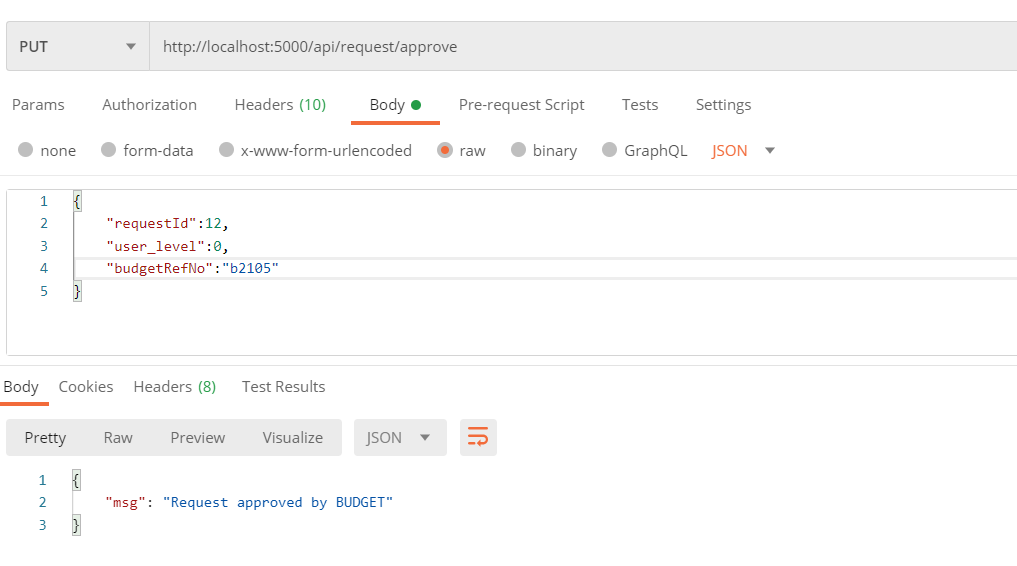


**Email notifications:**

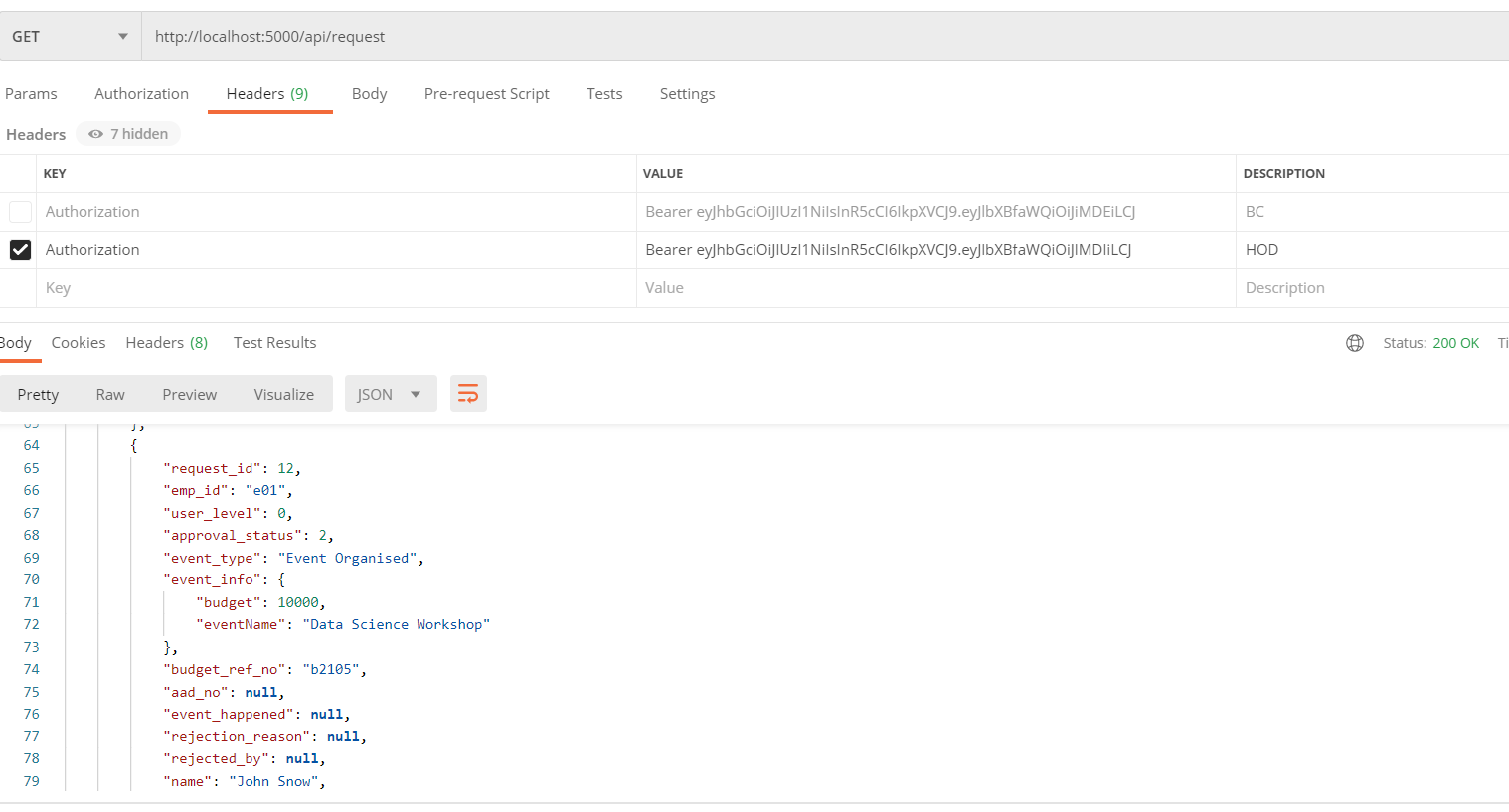


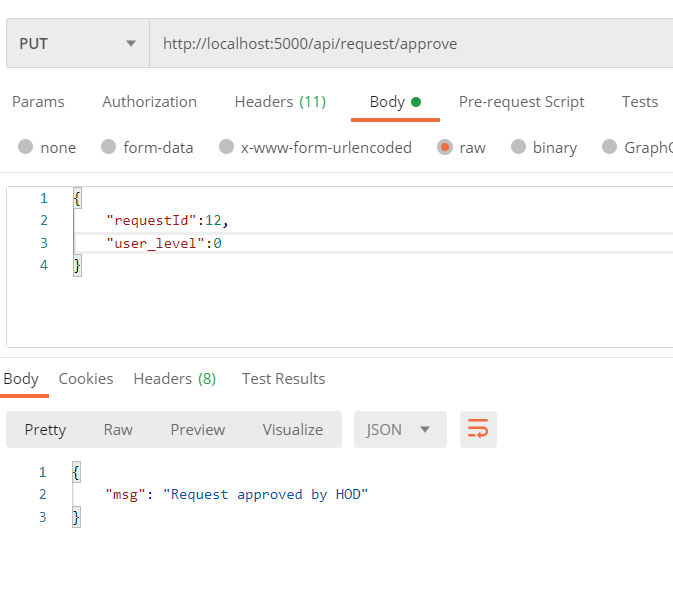


**Approving event as budget coordinator :**

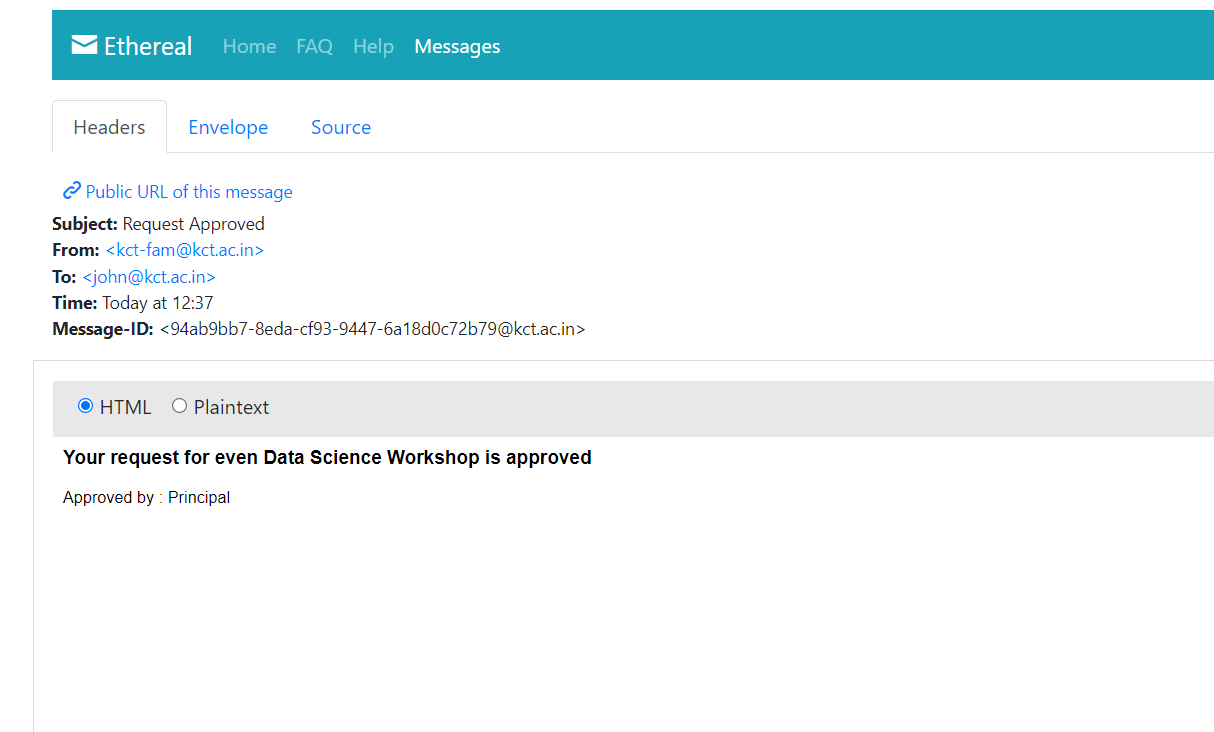


**Event approved and HOD received the request**

****

****

**After Completing all levels event will be created and acknowledgement mail will be sent to the users :**

****

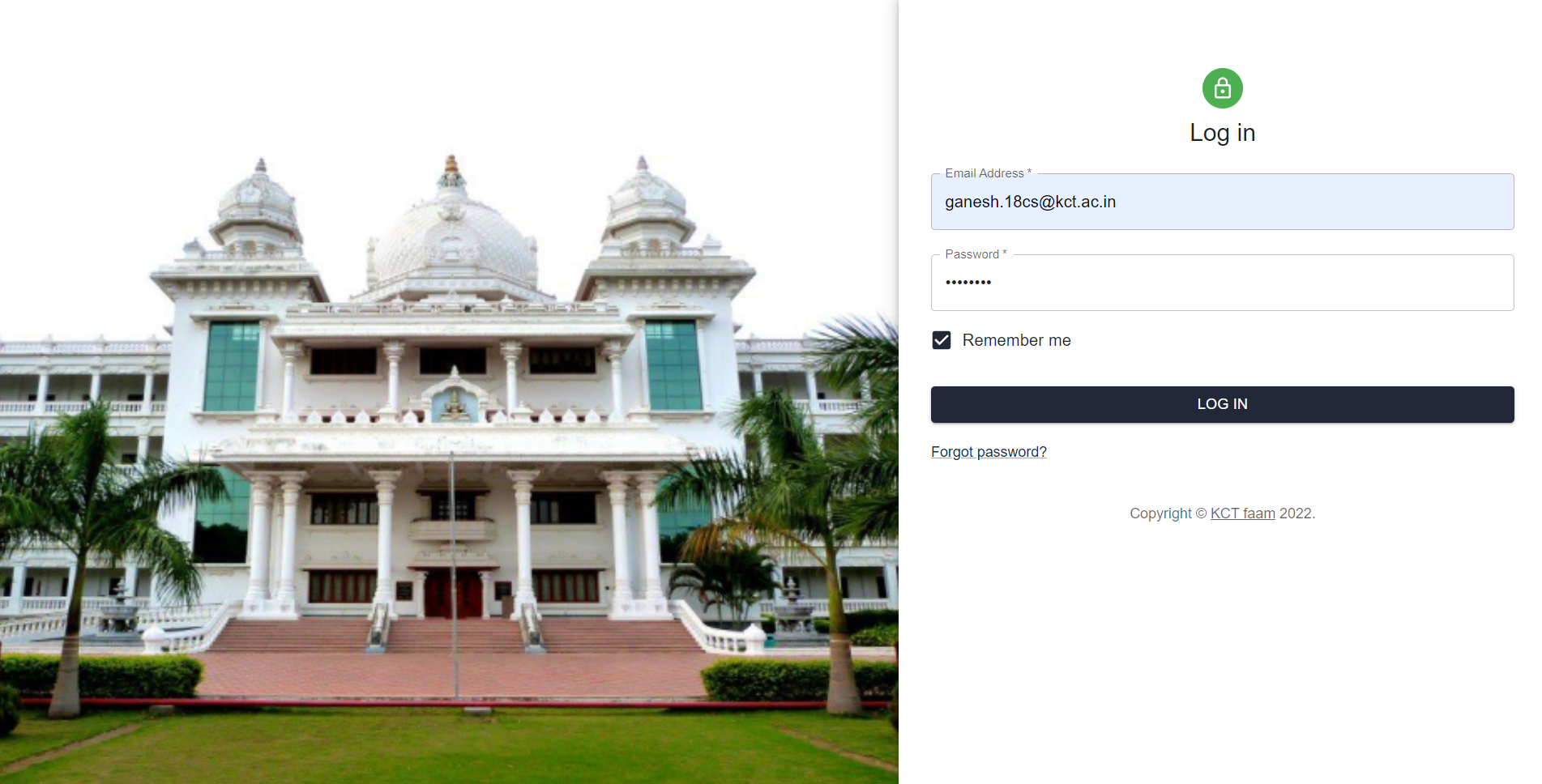
**UI/UX design and implementations:**

An interactive and user-friendly UI is designed and developed using modern javascript libraries like React js. React js uses a virtual dom which makes it faster and creates Single Page applications (SPA) more efficiently and effectively.

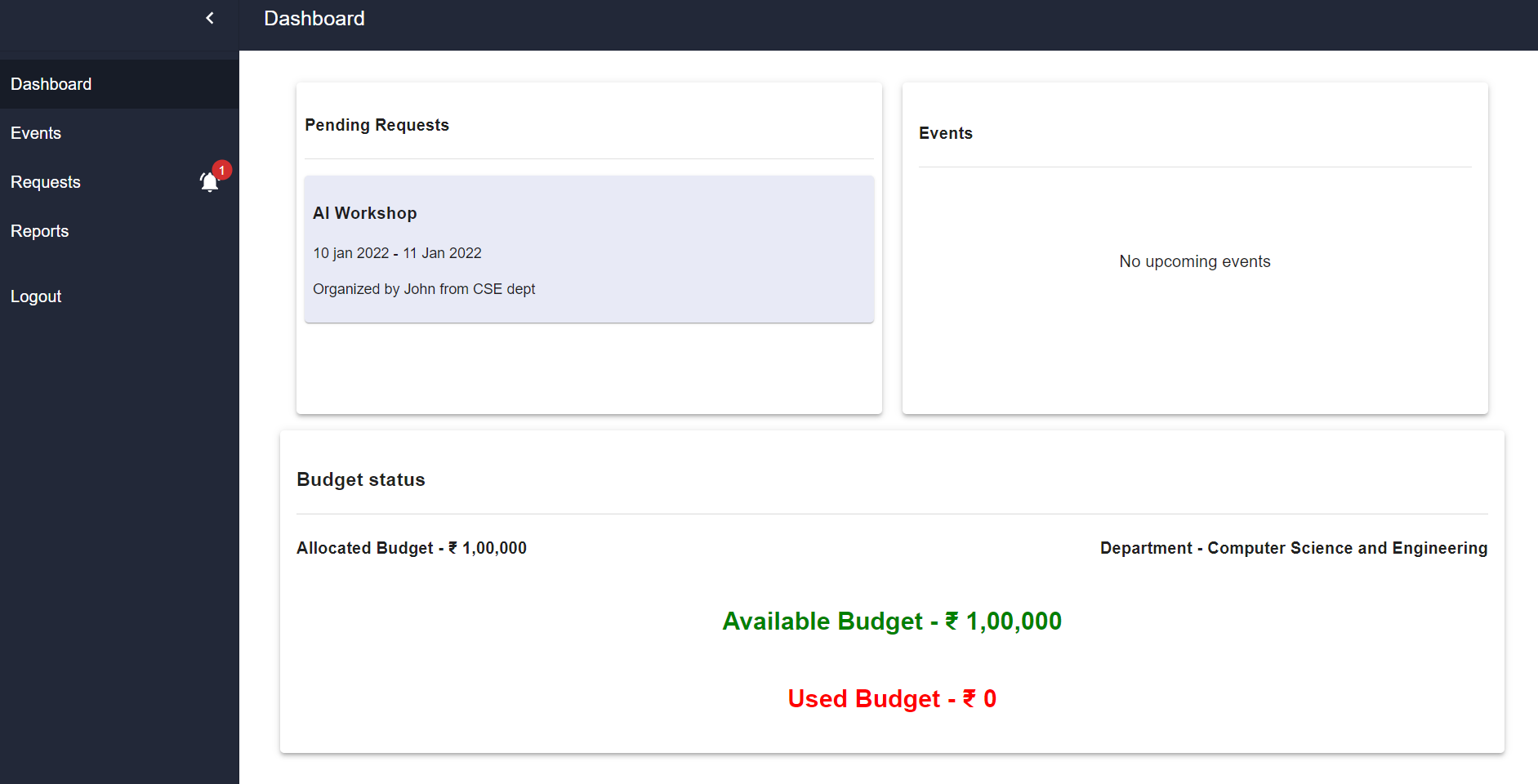
Redux - a state management tool is used along with react to manage, store and pass states of different components without repeatedly passing it as properties in every component.

**UI/UX:**

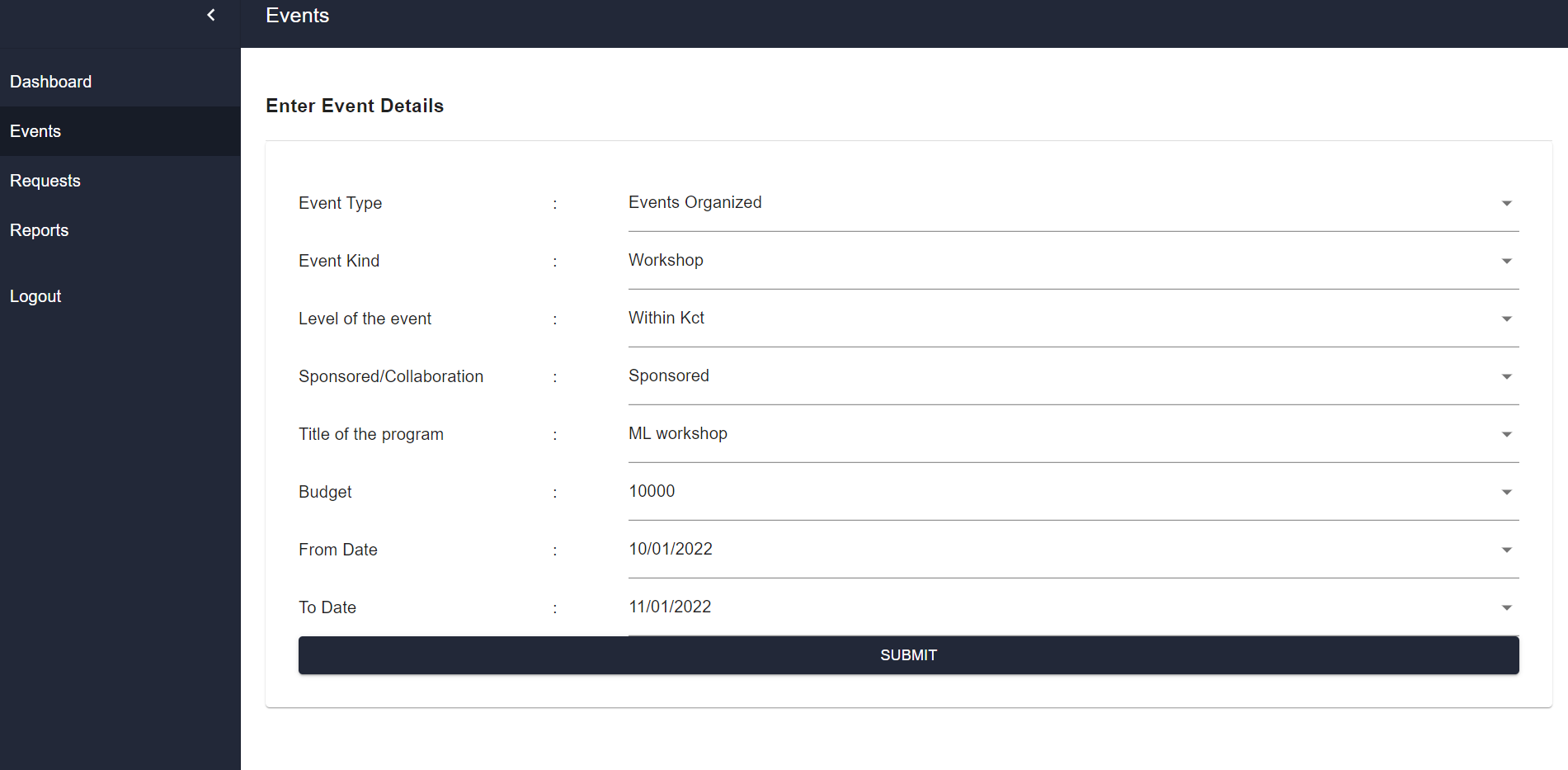
**Login page:**

****

**Dashboard:**

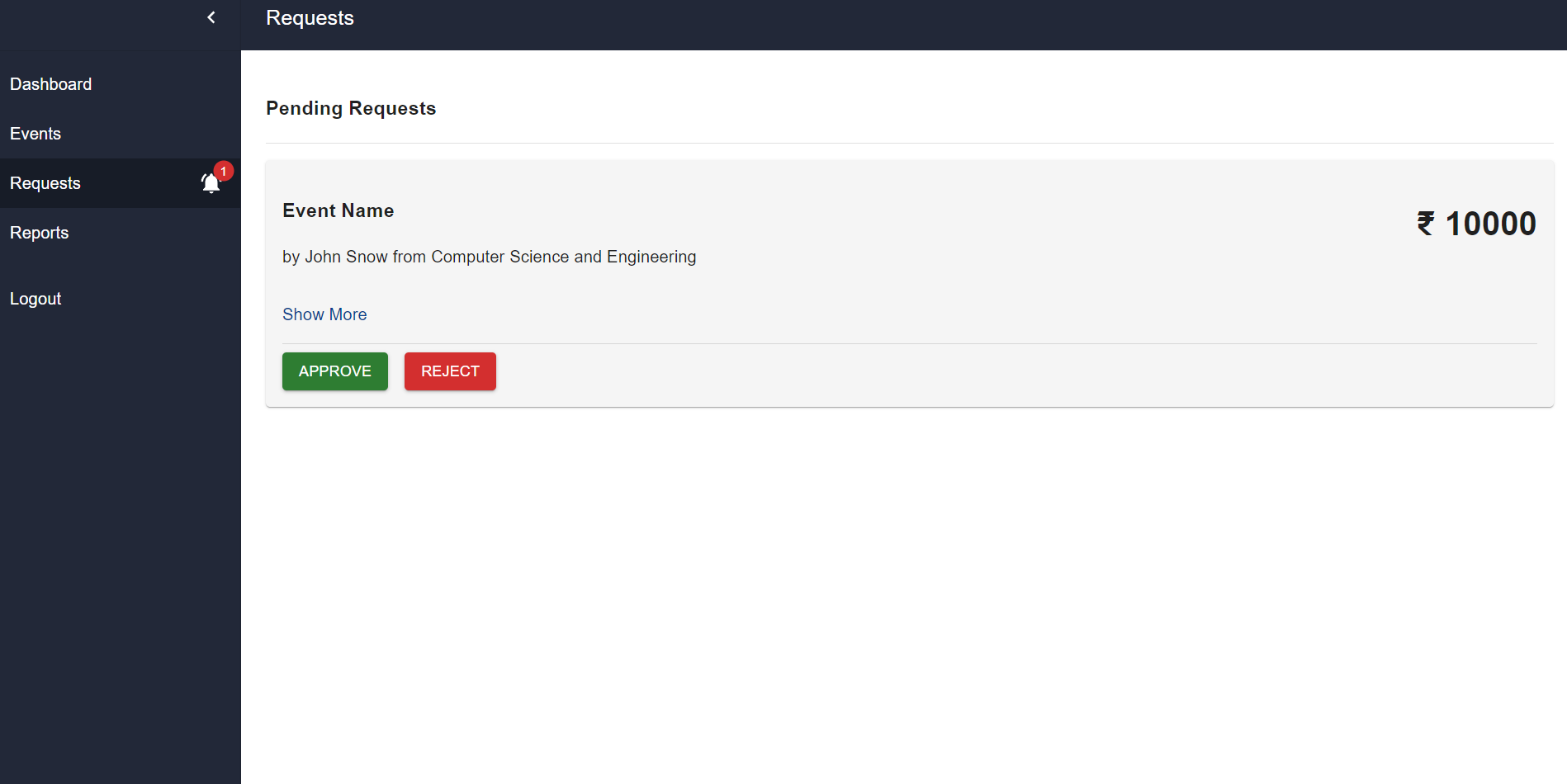
****

**New Event Form**

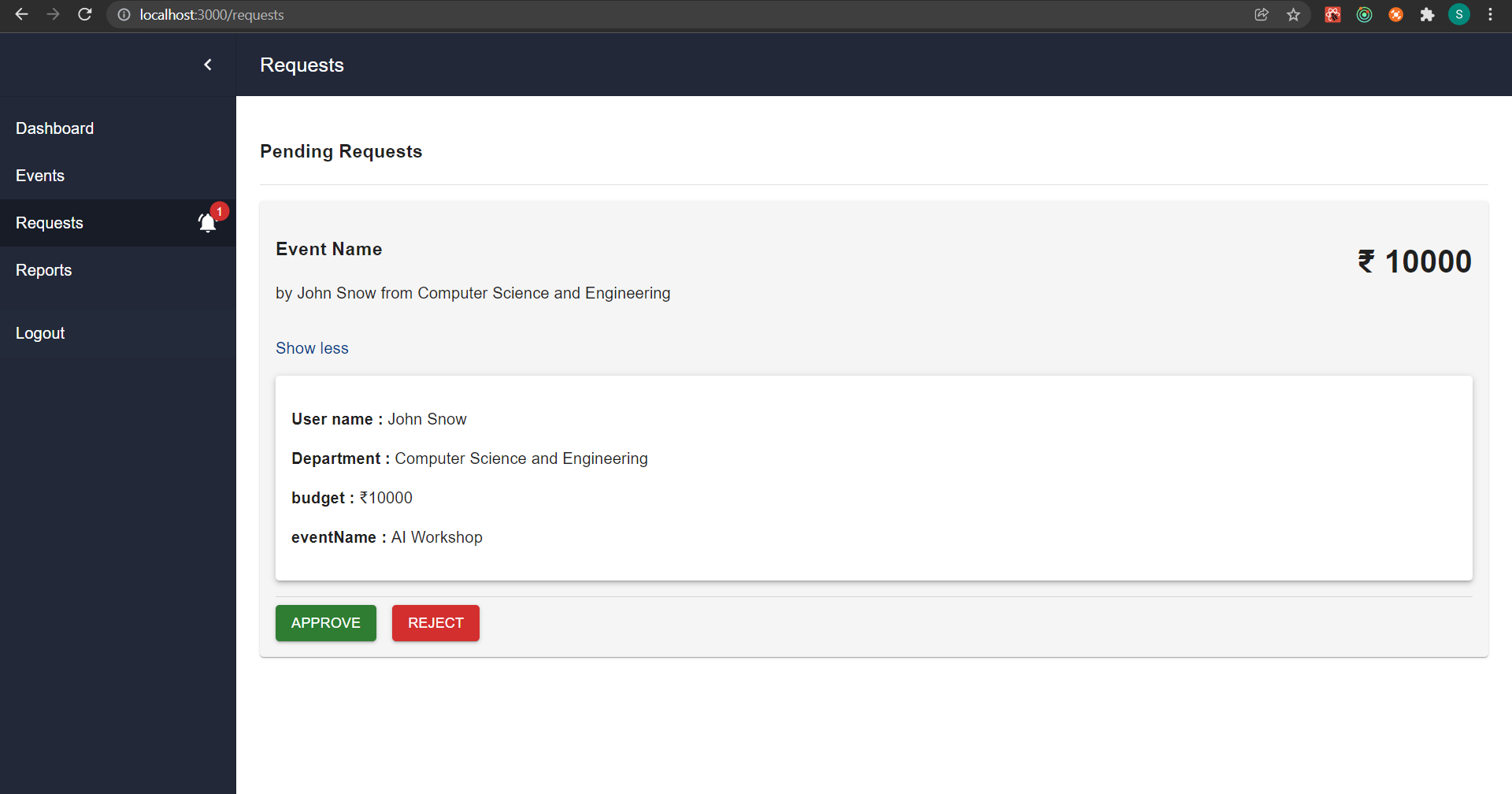
****

**View Request Page :**

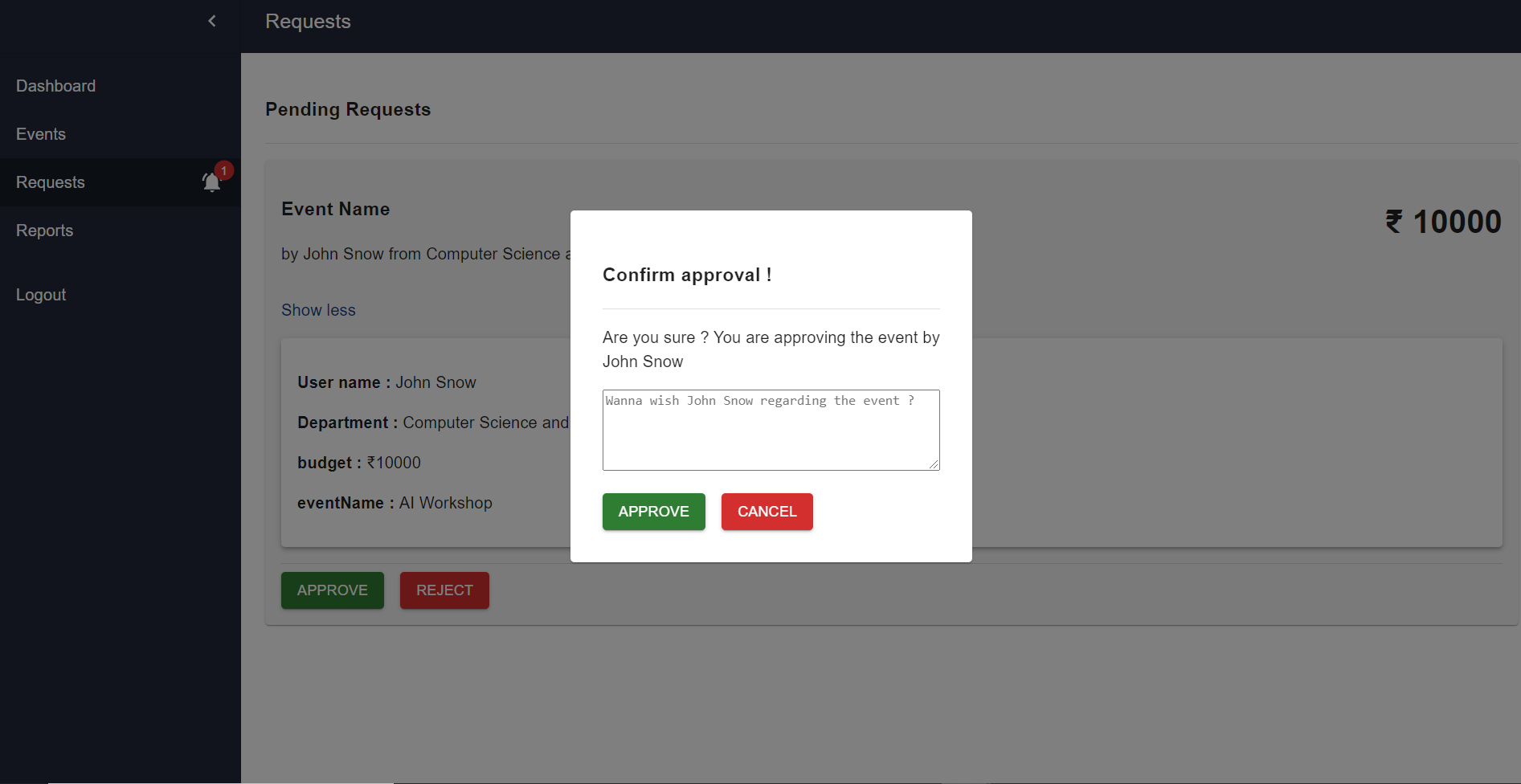
**feature to view the pending requests:**

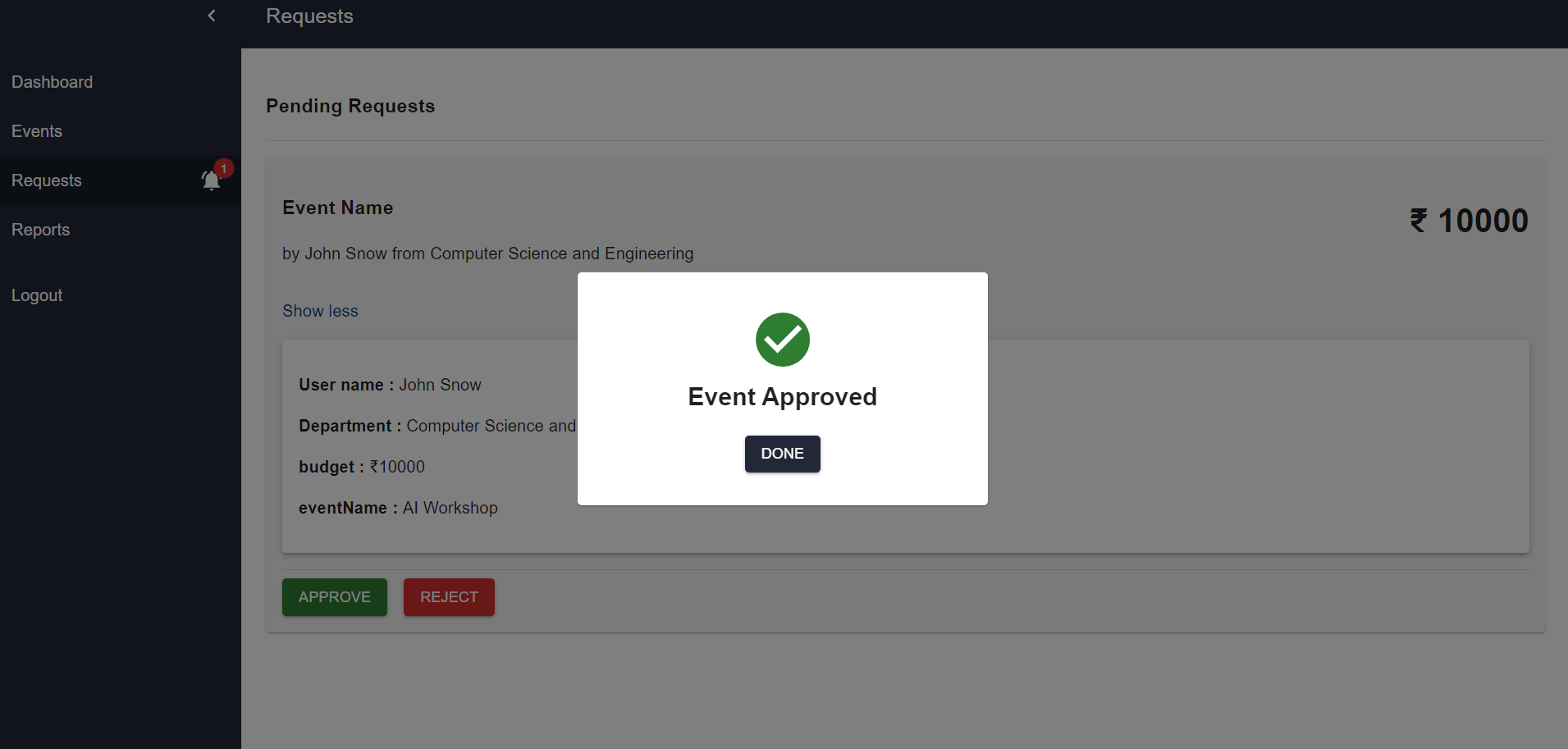
****

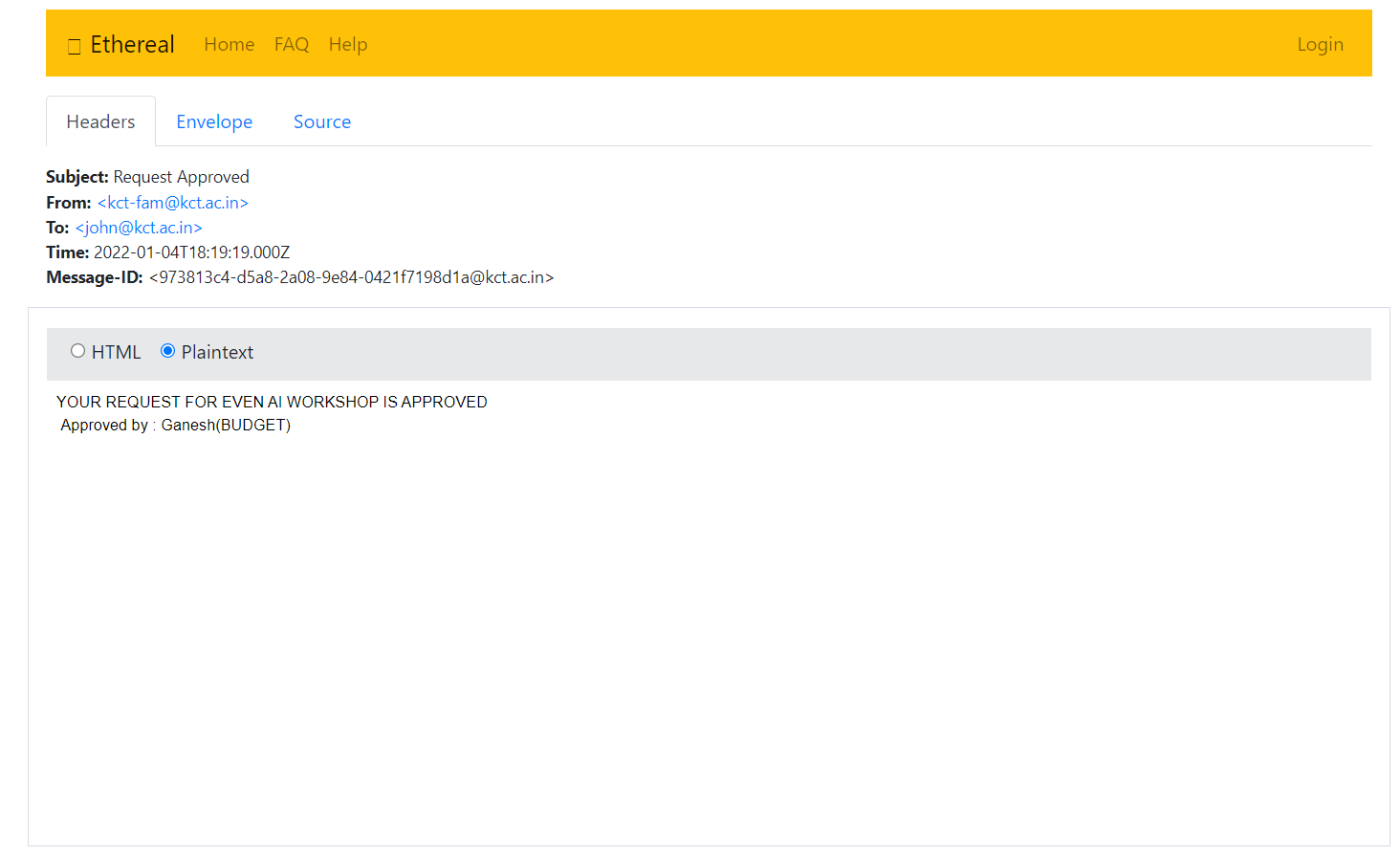
**Show more to view the details of the request:**

****

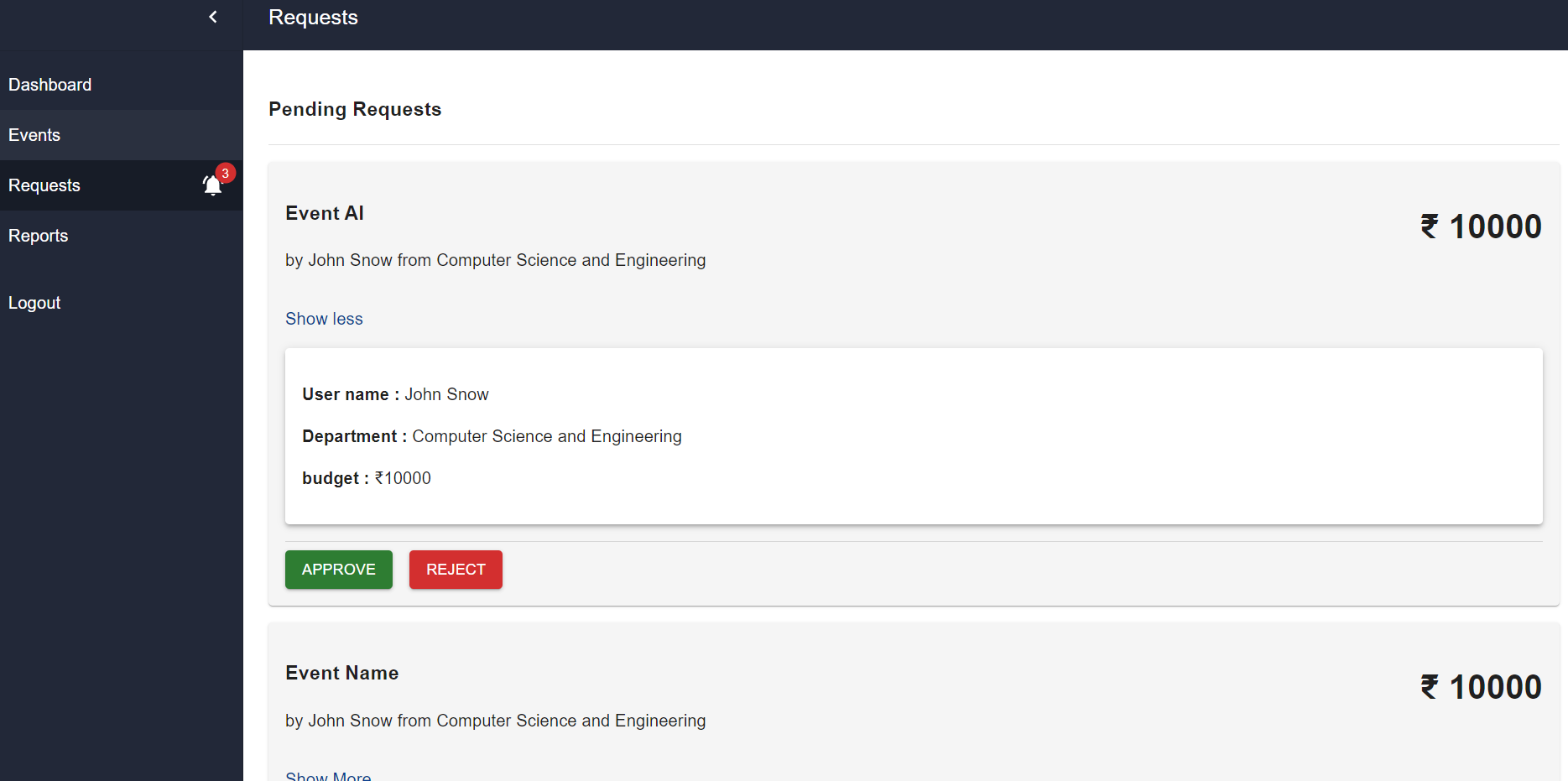
**Approve / Reject the request with a message to the user:**

****

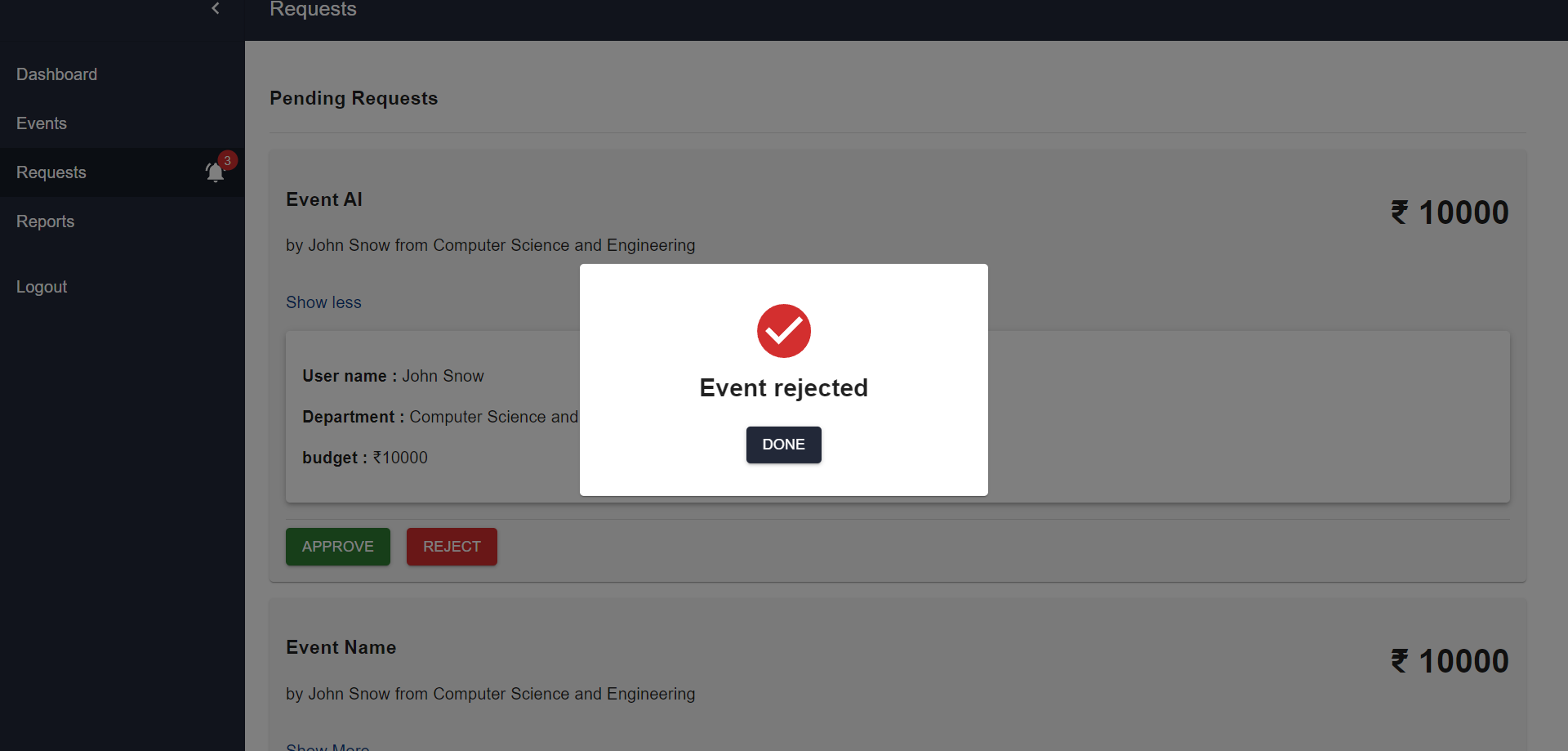
****

****

**Request moved to HOD :**

****

**Request Rejection:**

****

**5. SYSTEM REQUIREMENTS**

**5.1 USER SIDE REQUIREMENTS**

• Operating System - Windows or Linux or Mac

• Machine- Desktop or Laptop

• Web Browser - Chrome or Morzilla or Brave

• Internet Connection

• Mail Client - outlook/gmail

**6. CONCLUSION**

The quest to make life easier and processing faster has led to computerization of various processes. Computer technology has transformed so many sectors especially the educational sector in no small measure. In an effort to foster technology driven systems, a faculty archival and approval management system has been proposed. In conclusion, from proper analysis and assessment of the existing systems it can be safely concluded that the system needs various features and functionalities to be included. It is working properly and adequately meets the minimum expectations that were for it initially. The new system is expected to give benefits to the users and staff in terms of efficiency in the archival and approval activities.

**7. APPENDIX**

**REST API:**

*const express = require("express");*

*const app = express();*

*const db = require("./db/db");*

*const cors = require("cors");*

*const sendMail = require("./common/mail");*

*require("dotenv").config();*

*const PORT = process.env.SERVER\_PORT;*

*const corsOption = {*

*origin: process.env.FRONT\_END\_DOMAIN,*

*};*

*db.connect();*

*//midlewares*

*app.use(express.json());*

*app.use(cors(corsOption));*

*//routes*

*app.use("/api/admin", require("./routes/adminRoute"));*

*app.use("/api", require("./routes/authenticateRoute"));*

*app.use("/api/request", require("./routes/requestsRoute"));*

*app.use("/api/users", require("./routes/userRoute"));*

*app.get("/", async (req, res) => {*

*sendMail(*

*["chsubash333@gmail.com", "subash.18cs@kct.ac.in"],*

*"test real",*

*"hi hello"*

*)*

*.then(() => {*

*res.send({*

*msg: "welcome",*

*});*

*})*

*.catch((err) => res.send({ err }));*

*});*

*//server listen port*

*app.listen(PORT, () => {*

*console.log(`app running on port ${PORT} successfully`);*

*});*

**Database Configurations :**

*const mysql = require("mysql2");*

*require("dotenv").config();*

*const connection = mysql.createConnection({*

*host: process.env.DB\_HOST,*

*user: process.env.DB\_USER,*

*database: process.env.DB\_DATABASE,*

*});*

*module.exports = connection;*

**Request Route**

*const router = require("express").Router();*

*const db = require("../db/db");*

*const authenticate = require("../common/authenticate");*

*const {*

*findUserVal,*

*requestSentMail,*

*notifyMail,*

*findDesignation,*

*rejectMail,*

*approveMail,*

*} = require("../common/functions");*

*const {*

*NORMAL\_USER,*

*BUDGET\_COORDINATOR,*

*HOD,*

*HR,*

*ARCHIVE,*

*PRINCIPAL,*

*REJECTED,*

*APPROVED,*

*} = require("../common/constants");*

*router.post("/", authenticate.auth, (req, res) => {*

*let userLevel = findUserVal(req.user);*

*let nextUserLevel = 0;*

*let values = [*

*req.user.emp\_id,*

*userLevel,*

*req.body.eventType,*

*JSON.stringify(req.body.data),*

*];*

*let query = "";*

*if (userLevel === BUDGET\_COORDINATOR) {*

*query =*

*"INSERT INTO requests(emp\_id,user\_level,event\_type,event\_info,approval\_status,budget\_ref\_no) VALUES(?,?,?,?,?,?)";*

*values = [...values, HR, req.body.budgetRefNo];*

*nextUserLevel = HR;*

*} else if (userLevel === ARCHIVE) {*

*query =*

*"INSERT INTO requests(emp\_id,user\_level,event\_type,event\_info,approval\_status,aad\_no) VALUES(?,?,?,?,?,?)";*

*values = [...values, BUDGET\_COORDINATOR, req.body.aadNo];*

*nextUserLevel = BUDGET\_COORDINATOR;*

*} else {*

*query =*

*"INSERT INTO requests(emp\_id,user\_level,event\_type,event\_info,approval\_status) VALUES(?,?,?,?,?)";*

*values = [...values, BUDGET\_COORDINATOR];*

*nextUserLevel = BUDGET\_COORDINATOR;*

*}*

*db.query(query, values, (error, result, fields) => {*

*if (error) {*

*console.log(error);*

*res.status(500).send({*

*error,*

*});*

*} else if (result && result.affectedRows) {*

*res.send({*

*msg: "request sent successfully",*

*});*

*requestSentMail(req.user, {*

*...req.body.data,*

*eventType: req.body.eventType,*

*});*

*notifyMail(req.user, findDesignation(nextUserLevel), {*

*...req.body.data,*

*eventType: req.body.eventType,*

*});*

*} else {*

*console.log(result);*

*res.status(500).send({*

*result,*

*});*

*}*

*});*

*});*

*router.get("/", authenticate.auth, (req, res) => {*

*const userLevel = findUserVal(req.user);*

*const callBack = (error, result) => {*

*if (error) {*

*console.log(error);*

*res.status(500).send({*

*error,*

*});*

*} else if (result && result.length) {*

*res.send({*

*requests: [...result],*

*});*

*} else {*

*res.send({*

*msg: "No requests found",*

*});*

*}*

*};*

*if (userLevel === HOD) {*

*console.log(userLevel);*

*db.query(*

*"SELECT \* FROM requests \*

*join users u1 on requests.emp\_id=u1.emp\_id \*

*join departments d on d.department=u1.department where approval\_status=? and u1.department=?",*

*[userLevel, req.user.department],*

*callBack*

*);*

*} else {*

*db.query(*

*"SELECT \* FROM requests \*

*join users u1 on requests.emp\_id=u1.emp\_id \*

*join departments d on d.department=u1.department where approval\_status=? ",*

*userLevel,*

*callBack*

*);*

*}*

*});*

*router.put("/reject", authenticate.auth, (req, res) => {*

*const userLevel = findUserVal(req.user);*

*db.query("UPDATE requests SET approval\_status=?,rejected\_by=?,rejection\_reason=? WHERE approval\_status=? AND request\_id=?",*

*[*

*-1,*

*req.user.emp\_id,*

*req.body.rejectionReason,*

*userLevel,*

*req.body.requestId,*

*],*

*(error, result) => {*

*if (error) {*

*console.log(error);*

*res.status(500).send({*

*error,*

*});*

*} else if (result && result.affectedRows) {*

*res.send({*

*msg: "Request declined successfully",*

*});*

*rejectMail(req.body.requestId, req.user, req.body.rejectionReason);*

*} else if (result && !result.affectedRows && !result.fieldCount) {*

*res.send({*

*msg: "Row not found",*

*});*

*} else {*

*res.status(500).send({*

*result,*

*});*

*}*

*}*

*);*

*});*

*router.put("/approve", authenticate.auth, (req, res) => {*

*const userLevel = findUserVal(req.user);*

*const requestedUserLevel = req.body.userLevel;*

*let nextLevel = 0;*

*let values = [];*

*let query =*

*"UPDATE requests SET approval\_status=? WHERE approval\_status=? AND request\_id=?";*

*if (userLevel === BUDGET\_COORDINATOR) {*

*if (requestedUserLevel === NORMAL\_USER) nextLevel = HOD;*

*else if (requestedUserLevel === HOD) nextLevel = HR;*

*else if (requestedUserLevel === HR) nextLevel = ARCHIVE;*

*else if (requestedUserLevel === ARCHIVE) nextLevel = PRINCIPAL;*

*else if (requestedUserLevel === PRINCIPAL) nextLevel = APPROVED;*

*} else {*

*nextLevel = userLevel + 1;*

*}*

*values = [nextLevel, userLevel, req.body.requestId];*

*db.query(query, values, (error, result) => {*

*if (error) {*

*console.log(error);*

*res.status(500).send({*

*error,*

*});*

*} else if (result && result.affectedRows) {*

*res.send({*

*msg: "Request approved",*

*});*

*approveMail(req.body.requestId, req.user);*

*} else {*

*console.log(query);*

*console.log(values);*

*res.status(500).send({*

*result,*

*}); } });});*

*module.exports = router;*

**Github Link:**

<https://github.com/ganesh-sethu/kct-fam-app>

**References:**

[1] Trygve Reenska, http:// heim.ifi.uio.no / ~trygver/themes/mvc/mvc-index.html.

[2] Glenn E. Krasner, Stephen T. Pope, “A cookbook for using the model-view controller user interface paradigm in Smalltalk-80”, Journal of Object-Oriented Programming, vol. 1, no. 3, 1988, pp. 26-49.

[3] Kevin McArthur, Pro PHP: Patterns, Frameworks, Testing and More, Apress, 2008.

[4] A Freeman, S Sanderson, Pro ASP.NET MVC 3 Framework, Apress, 2011.

[5] W. J. Gilmore, Easy PHP Websites, Columbus, Ohio: W.J. Gilmore, LLC, 2009.

[6] R. Battle and E. Benson, “Bridging the semantic Web and Web 2.0 with Representational State Transfer (REST),” Web Semantics.Sci . Serv. Agents World Wide Web, vol. 6, no. 1, pp. 61–69, Feb. 2008.

[7] B.Costa, P. F. Pires, F. C. Delicato, and P. Merson, “Evaluating a Representational State Transfer (REST) Architecture: What is the Impact of REST in My Architecture?,” in 2014 IEEE/IFIP Confer-ence on Software Architecture, 2014, pp. 105–114.

[8] G. Schermann, J. Cito, and P. Leitner, “All the Services Large and Micro: Revisiting Industrial Practice in Services Computing,” in Service-Oriented Computing – ICSOC 2015 Workshops, 2015, pp. 36–47.

[9] C. Rodríguez et al., “REST APIs: A Large-Scale Analysis of Compliance with Principles and Best Practices,” in Web Engineering,vol. 9671, A. Bozzon, P.Cudre-Mauroux, and C. Pautasso, Eds. Cham: Springer International Publishing, 2016, pp. 21–39.

[10] F. Haupt , F. Leymann , A .Scherer,and K .Vukojevic-Haupt, “A Framework for the Structural Analysis of REST APIs,” in Software Architecture (ICSA), 2017 IEEE International Conference on, 2017, pp. 55–58.