

1.INTRODUCTION

“IT FEST MANAGEMENT SYSTEM” is a Python based project. IT Fest Management System is a software developed for fully computerised and online based IT fest management. This software solution help students from various colleges to register through online. This can be used by the organization to manage the events and maintain the records. The IT Fest Management system provide a user-friendly interface easy operation. The software automates the processes that were done manually and that were developed as a partial automated one, thus eliminating the time taken for various activities. The system is developed as a user friendly. The main highlights of this system are less time consuming, less human resource, flexible and efficient.

2.SYSTEM ANALYSIS

System analysis is a detailed study of the various operations performed by the system and their relationship within and module of the system. It is a structured method for solving the problems related to the development of a new system. The detailed investigation of the present system is the focal point of system analysis. This phase involves the study of parent system and identification of system objectives. Information has to be collected from all people who are affected by or who use the system. During analysis, data are collected on the variable files, decision point and transactions handled by the present system. The main aim of system is to provide the efficient and user-friendly automation. So, the system analysis process should be performed with extreme precision, so that an accurate picture of existing system, its disadvantages and the requirements of the new system can be obtained.

In this phase, the problem is identified and an alternate system solution for solving it is recognized. System Analysis not only includes the process of synthesis, which is a process of putting parts together to form a new product, but also the requirement identification and specification. The System Analysis include the studying of the existing system and its drawback, designing a new system and conducting cost benefit analysis. The costs and benefits of each alternative guide the selection of the best system for the job. System analysis is a problem-solving activity that requires intensive communication between the system user and the system developer.

2.1 EXISTING SYSTEM

The current management system is manual. Data are collected and maintained manually. Participant registration is done through by contacting the coordinator or by directly visiting them. So, it takes time. Time schedule for each program is listed in the notice board and every updating must be done in the board and known to the participants and others only through visiting the notice board periodically. So, if any changes have to be done it must be reached to everyone and it is difficult in manual management system.

LIMITATION OF EXISTING SYSTEM

- The existing IT fest management system is not computerized.
- It does not provide any online facilities.
- There is a difficulty for data maintenance.
- Difficulty in event scheduling.
- Registration is more time-consuming process.
- More manpower required.

2.2 PROPOSED SYSTEM

The proposed system is to computerize all the manual activities in the existing system.

ADVANTAGES OF PROPOSED SYSTEM

- This software's prime objective is to provide online registration for candidates from various college.
- Coordinators can schedule the events and provide accommodation facilities based on the registration details.
- Provide information about timing, events and result on the site.

2.3 System Requirement Specification (SRS)

2.3.1 Software Specification

Platform : Windows

Front end : Python

Scripting Language : JavaScript, HTML

Department of Computer Science, Nirmala College, Muvattupuzha

Back end : MySQL

IDE : Visual Studio Code

2.3.2 Hardware Specification

Microprocessor : Any 64 bit processor

Speed : 2.44 GHz

RAM : 256 MB and above

Hard Disk : 40 GB and above

Key board : 108 key standard key board

Mouse : Standard Mouse

Device : Serial Communication Device

2.3.3 Front End: PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging

Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

2.3.4 Back End: MySQL

MySQL is the world's most popular open-source database. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo! and many more. Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded applications. They differ in the scope of services and in price. Additionally, a number of third-party organizations exist to provide support and services, including Maria DB and Persona.

2.4 FEASIBILITY ANALYSIS

Feasibility study is carried out to select the best system that meets performance requirements. This entails identification, description, and evaluation of candidate systems and selection of the best system for the job. The main aim of the feasibility study is to evaluate alternative system and to propose the most feasible and desirable system for development. During the detailed study conducted in existing system, I have noticed several problems. It is a time consuming and a job pending one. After Initial investigation culminates in a proposal that determines whether an alternative system is feasible.

Mainly there are four types of feasibility analysis:

- Economic Feasibility
- Technical Feasibility

- Behavioural Feasibility
- Operational Feasibility

Economic Feasibility

Economic feasibility analysis is also known as cost/benefit analysis. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. The proposed system reduces the operating cost in terms of time by automating the process. This system is economically feasible.

Technical Feasibility

Technical feasibility centres on the existing system and to what extent it can support the proposed addition. This system must be evaluated from technical view point first. The assessment of this feasibility must base on the outline design of the system requirements in the terms of inputs, outputs program procedure. The minimum requirement for this proposed system is a system. The existing system has this entire requirement. So, the proposed system is said to be technically feasible.

Behavioural Feasibility

People are inherently resistant to changes and computer is known for facilitating the changes. An estimate should be made to how strongly the users react towards the development of the system. The proposed system consumes less time. Thus, the people are made to engage in some other important work.

Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

2.5 DATAFLOW DIAGRAM

2.5.1 Introduction

A graphical representation is used to describe and analyse the movement of data through a system manual or automated including the processes, storing of data and delays in system. Data flow diagrams are central tool and the basis from which other components are developed. The transformation of data, from input to output through process maybe described logically and independently of the physical components associated with the system. They are termed logical data flow diagrams, showing the actual implementation and the movement of data between people, departments and workstations. DFD is one of the most important modelling tools used in system designed shows the flow of data through different process in system purpose.

Throughout my project, the context flow diagram, flow charts have been extensively used to achieve the successful design of the system. The efficient design of the dataflow and context flow diagram helps to design the system successfully without much major flows within the scheduled time. This is the most complicated part in project. The designing process, my project took more than the activities in the software life cycle. Structured analysis is a set of techniques and graphical tools that help the analyst to develop a new kind of system specification that are easily understandable to the user.

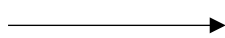
DFD's show the major decompositions of the system functions and their interfaces. The DFD is graphic and presents a picture of what is being specified and is conceptually easy to understand presentation of the application.

One important feature of DFD's is that it is logical rather than physical. The elements of the system do not depend on vendor or hardware. They specify in precise, concise manner the working of the system and how it hangs together.

DFD is the graphic representation of data movement process, and files used in support of an information system. There are several rules of thumb used in drawing DFDs.

- Process should be named and numbered for easy references.
- The direction of flow is from top to bottom and from left to right.
- When a process is imported in the lower-level details, they must be numbered.
- Process and data flow names have the first letter of each word must be a capital letter

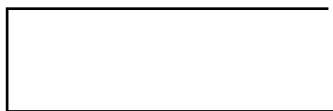
The four main symbols used for developing a DFD are shown below:



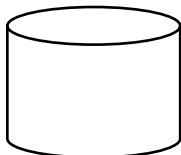
Data flow



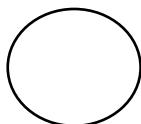
Data Source or Destination



Table



Data Store

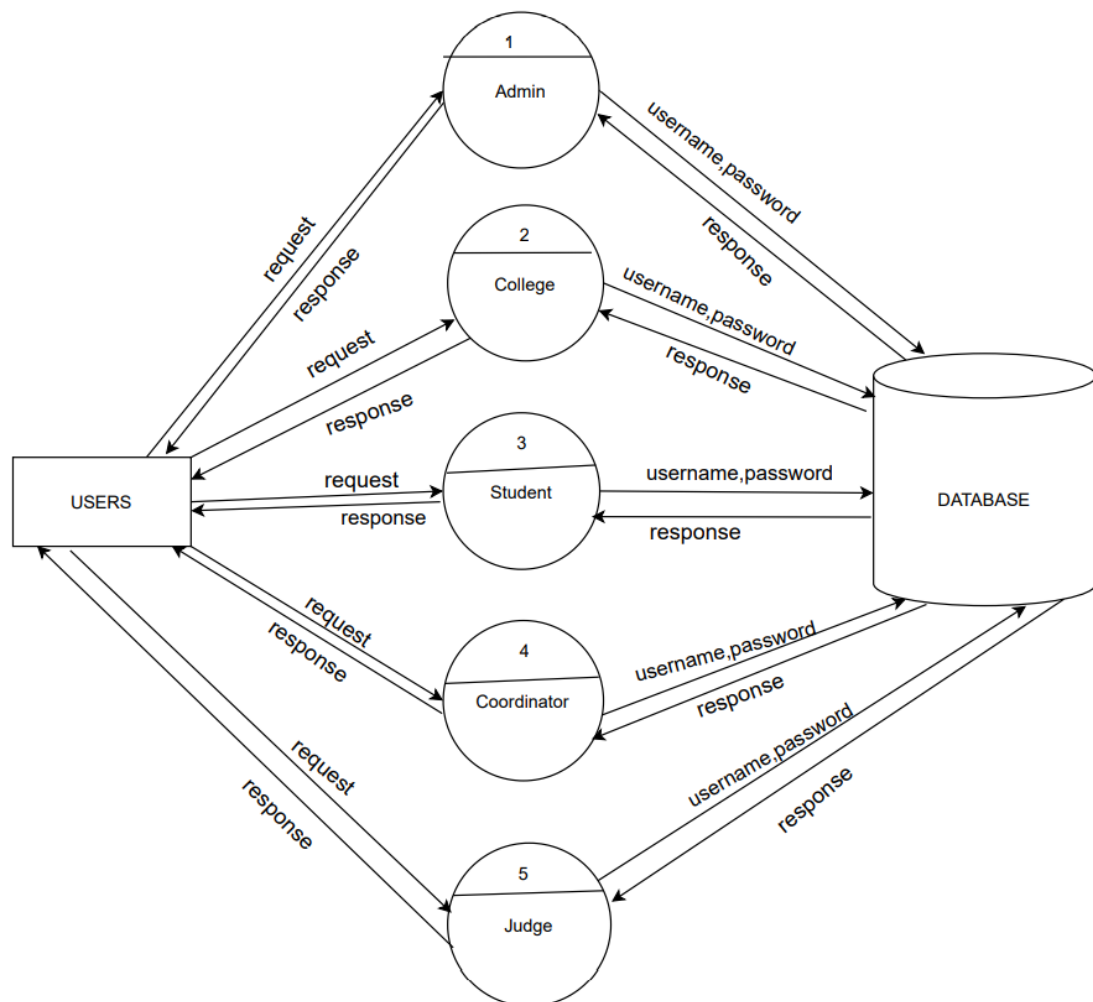


Represents a process that transforms data streams

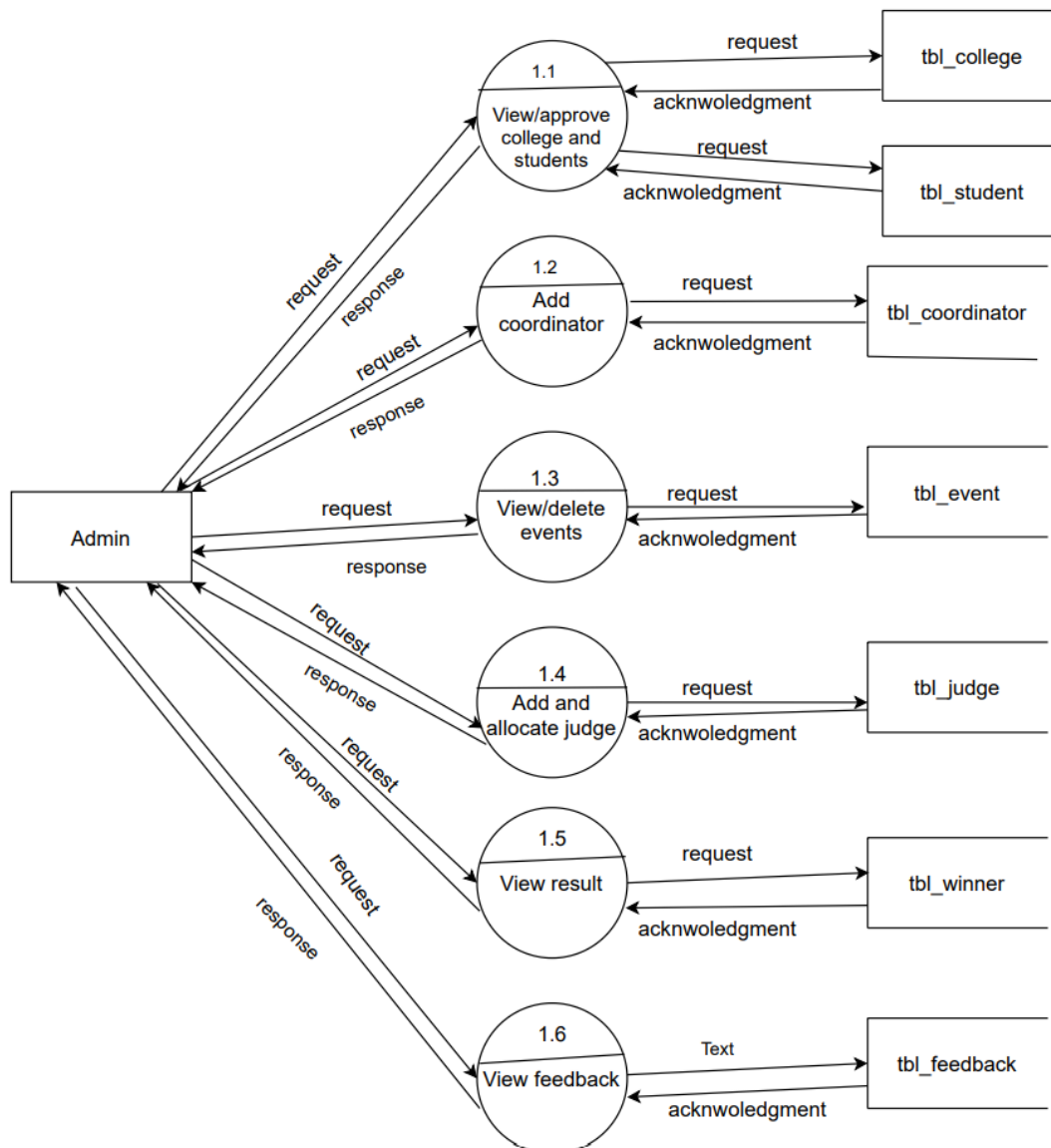
LEVEL 0 DFD

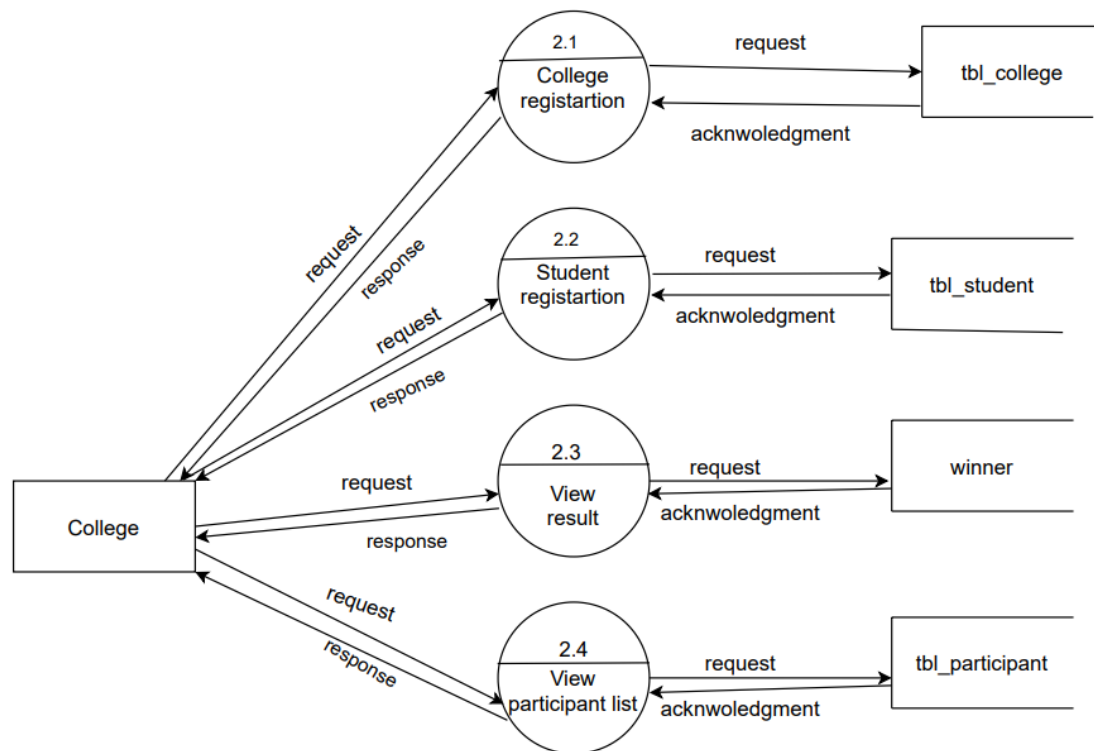


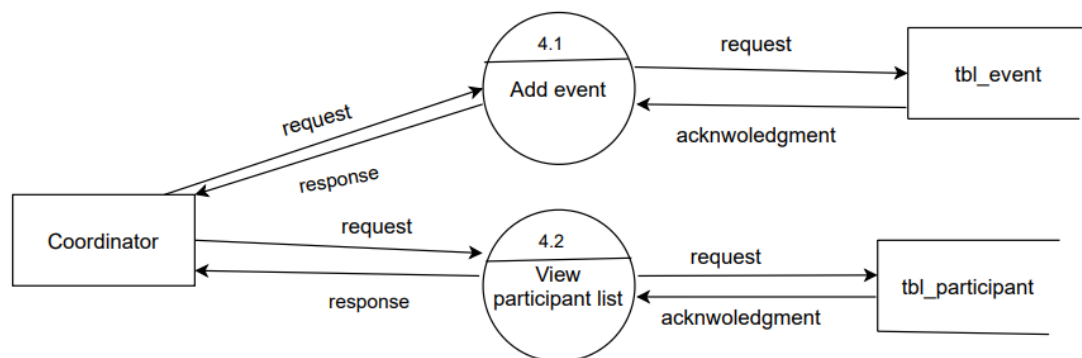
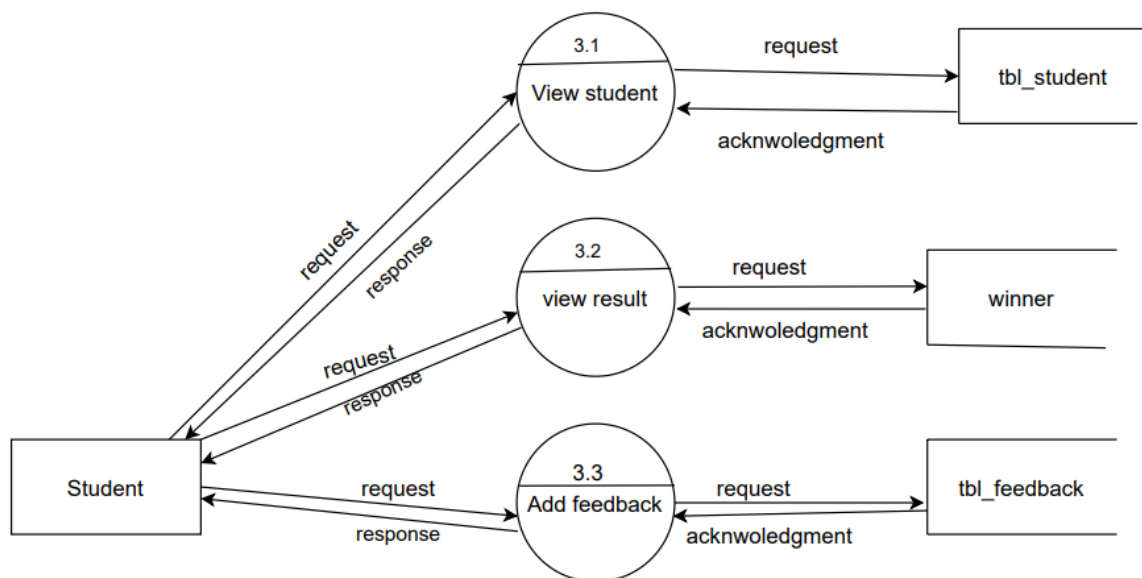
LEVEL 1 DFD

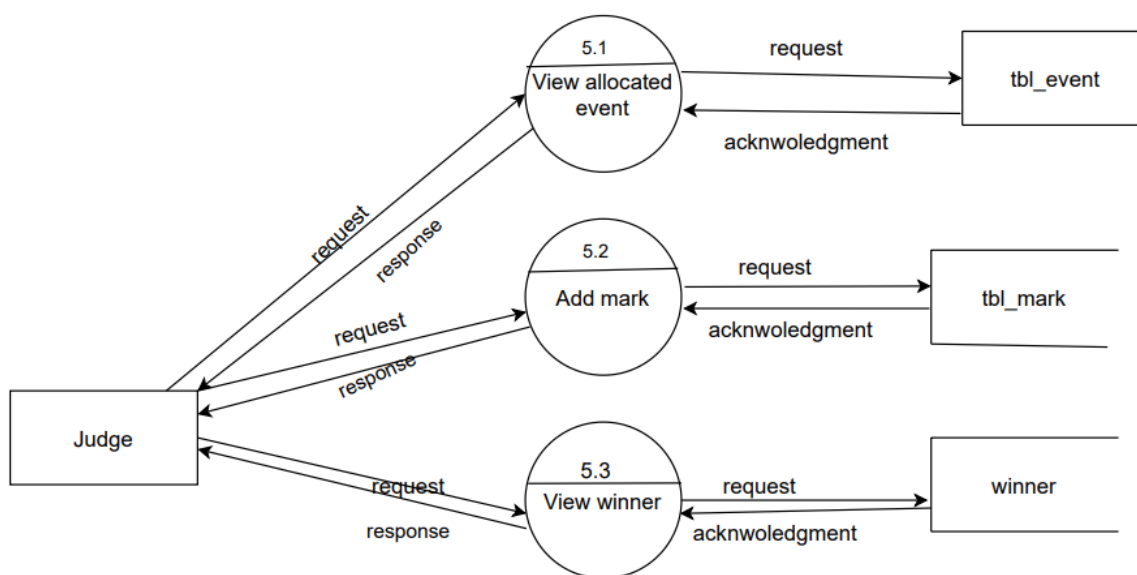


LEVEL 2 DFD









3.SYSTEM DESIGN

System design is the creative act of invention developing new inputs, a database offline files, method, procedures and output for processing business data to meet organization objectives. The design phase focuses on the detailed implementation of the system recommended in the feasibility study. The design phase is a transition from user-oriented document to a document oriented to the programmers or database personnel. Characteristics of a well-designed system are:

- Accessibility
- Decision making ability
- Economy
- Flexibility
- Reliability
- Simplicity

The design will determine the success of the system. System design is based on the information gathered during system analysis. System design goes through two phases of development.

- Logical Design –DFD shows the logical flow of a system and defines the boundaries of the system. For the candidate system it describes the inputs, outputs, databases and procedures-all in a format that meets the user's requirements.
- Physical Design - This produces the working system by defining the design specification that tells programmers exactly what the candidate system must do.

3.1 MODULE DESCRIPTION

This system describes the following modules:

Registration

The registration module is used to store the information's of the candidates and collages.

Login module

This module is used to enter the system for all users.

Coordinator module

The coordinator is adding the events and time schedule to the system

Judge module

The judge is assigned by the administrator. The judge can view the candidate who participate the events and enter score for the candidate

Company module

It companies can view the winners of the it fest and companies can select candidates for recruitment.

Admin module

The admin has all permission. The administrator is accept the collages wish to participate on it fest. The admin can also view the feedbacks from the collages and students. The admin can view winners

College module

The collages can add students who wish to participate the IT fest. Collages can send username and password to students through SMS. The collages can view result.

Student Module

Students can participate events and view result and also send feedbacks

3.3 DATABASE DESIGN

The data design transforms the information domain model created during analysis into the data structures that will be required to implement the software. The data objects and relationships defined in the entity relationship diagram and the detailed data content depicted in the data dictionary provide the basis for the data design activity. The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. Database Management System allows data to be protected and

organized separately from other resources. Database is an integrated collection of data. This is the difference between logical and physical data.

The organization of data in the database aims to achieve three major objectives:

- Data integration
- Data integrity
- Data independence

The databases are implemented using a DBMS package. Each particular DBMS has unique characteristics and general techniques for database design. There are 6 major steps in design process. The first 5 steps are usually done on paper and finally the design is implemented.

- Identify the table and relationships
- Identify the data that is needed for each table and relationship.
- Resolve the relationship
- Verify the design
- Implement the design

The database uses tables for storage. A table also contains records, which is a set of fields. All records, in a table have the same set of fields with different information. Each table contains key fields that establish relationships in the database and how the records are stored. There are primary key fields that uniquely identify a record in a table. There are also fields that contain the primary key from another table called foreign keys.

The various database tables that are used in this project are the following:

TABLE DESIGN**Table name: tblallocation**

Primary key: id

Foreign key: eid, jid

Field	Datatype	Constraint	Description
id	int	Primary Key, Auto increment	allocation id
eventid	int	Foreign key	event id
judgeid	int	Foreign key	judge id

Table name: tblcordinator

Primary key: cid

Field	Datatype	Constraint	Description
cid	int	Primary Key, Auto increment	cordinator id
name	varchar (50)	Not null	name
contact	varchar (50)	Not null	contact
place	varchar (50)	Not null	place
email	varchar (50)	Not null	email
status	varchar (50)	Not null	status

Table name: tblevent

Primary key: eid

Field	Datatype	Constraint	Description
eid	int	Primary Key, Auto increment	event id
ename	varchar (50)	Not null	event name
edate	date	Not null	date of event
venue	varchar (50)	Not null	venue
etime	time	Not null	time of event
Peoplenum	int	Not null	Number of people
status	int	Not null	

Table name: tblfeedback

Primary key: fid

Foreign key: sid

Field	Datatype	Constraint	Description
fid	int	Primary Key, Auto increment	feedback id
studentid	int	Foreign key	student id
feedback	varchar (50)	Not null	feedback

Table name: tbljudge

Primary key: jid

Field	Datatype	Constraint	Description
jid	int	Primary Key, Auto increment	judge id
name	varchar (50)	Not null	name
image	varchar (100)	Not null	To store image
contact	varchar (50)	Not null	contact
place	varchar (50)	Not null	place
email	varchar (50)	Not null	email
status	varchar (50)	Not null	status

Table name: tblstudent

Primary key: sid

Field	Datatype	Constraint	Description
sid	int	Primary Key, Auto increment	student id

sname	varchar (50)	Not null	name
saddress	varchar (50)	Not null	address
scontact	varchar (50)	Not null	contact
Splace	varchar (50)	Not null	place
semail	varchar (50)	Not null	email
status	varchar (50)	Not null	status

Table name: tblmark

Primary key: mid

Foreign key: pid, judgeid

Field	Datatype	Constraint	Description
mid	int	Primary Key, Auto increment	markid
pid	int	Foreign key	participant id
eventname	varchar (50)	Not null	event name
jid	int	Foreign key	judgeid
score	int	Not null	score

Table name: tblparticipants

Primary key: pid

Foreign key: mid, eid

Field	Datatype	Constraint	Description
mid	int	Foreign key	markid
pid	int	Primary key, Auto increment	participant id
eventid	int	Foreign key	event id
efile	varchar (100)	Not null	file

Table name: winner

Primary key: wid

Foreign key: mid, pid

Field	Datatype	Constraint	Description
wid	int	Primary key, Auto increment	Winner id
mid	int	Foreign Key	markid
pid	int	Foreign key	participant id
ename	varchar (50)	Not null	event name
score	int	Not null	score

Table name: College

Primary key: clgid

Field	Datatype	Constraint	Description
clgid	int	Primary key, Auto increment	To store college id
cname	Varchar(20)	Not null	To store college name
caddress	Bigint(12)	Not null	To store college address

ccontact	Varchar(20)	Not null	To store contact number
cemail	Varchar(20)	Not null	To store email address
status	Varchar(20)	Not null	

Table name:login_details

Name	Datatype	Constraint	Description
id	int	Primary key	To store id
username	Varchar(50)	Not null	To store username
password	Varchar(50)	Not null	To store password
usertype	Varchar(50)	Not null	To store usertype
status	Varchar(50)	Not null	

3.5 OUTPUT DESIGN

One of the important features of an information system for users is the output produces. Output is the information delivered to users through the information system. Without quality of the output, the entire system appears to be unnecessary that users will avoid using it. Users generally merit the system solely by its output. In order to create the most useful output possible. One works closely with the user though an interactive process. until the result is considered to be satisfactory.

4.SYSTEM TESTING

Testing is the process of examining the software to compare the actual behaviour with that of the expected behaviour. The major goal of software testing is to demonstrate that faults are not present. In order to achieve this goal, the tester executes the program with the intent of finding errors. Though testing cannot show absence of errors but by not showing their presence it is considered that these are not present. System testing is defined as the process by which one detects the defects in the software. Any software development organization or team has to perform several processes. Software testing is one among them. It is the final opportunity of any programmer to detect and rectify any defects that may have appeared during the software development stage. Testing is a process of testing a program with the explicit intension of finding errors that make the program fail. In short system testing and quality assurance is a review in software products and related documentation for completion, correctness, reliability and maintainability.

System testing is the first stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. testing is vital to the success of the system. System testing make a logical assumption that if all the parts of the system are correct and the goal will be successfully achieved. A series of testing are performed for the proposed system before the proposed system is ready for user acceptance testing.

The testing steps are:

- Unit testing
- Integration testing
- Validation
- Output testing
- Acceptance testing

System testing provides the file assurance that software once validated must be combined with all other system elements. System testing verifies whether all elements have been combined

properly and that overall system function and performance is achieved. After the integration of modules, the validation test was carried out over the system. It was found that all the modules work well together and meet the overall system function and performance. **a) Unit Testing**

Unit testing is carried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to located errors. This has enabled the detection of errors in coding and logic. Various test cases are prepared. For each module these test cases are implemented and it is checked whether the module is executed as per the requirements and outputs the desired result. In this test each service input and output parameters are checked. In unit testing

- Module interface was tested to ensure that information properly flows into and out of the program undertest.
- Boundary condition was tested to ensure that module operates properly at boundaries established to limit or restrict processing.
- All independent parts through the control structures were executed to ensure that all statements in the modules have been executed at least once.
- Error handling paths were also tested.

b) Integration testing

Integration testing is a systematic technique for constructing the program structure while the same time conducting test to uncover errors associated with interfacing. Unit tested module were taken and a single program structure was built that has been dictated by the design. Incremental integration has been adopted here. The modules are tested separately for accuracy and modules are integrated together using bottom-up integration i.e., by integrating from moving from bottom to the top. The system is checked and errors found during integration and rectified. In this testing, all the individual modules where combined and the module wise shifting was verified to be alright. The entire software was developed and tested in small segments, where errors were easy to locate and rectify. Program builds (group of

modules) were constructed corresponding to the successful testing of user interaction, data manipulation analysis and display processing and database management.

c) Validation Testing

Validation testing is done to ensure complete assembly of the error-free software. Validation can be termed successful only if it functions in manner. Reasonably expected by the customer under validation is alpha and beta testing. The customer-side validation is done in the testing phase. It is checked whether the data passed to each service is valid or not. Entering incorrect values does the validation testing and it is checked whether the errors are being considered. Incorrect values are to be discarded. The errors are rectified.

d) Output testing

After performing the validation testing the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen was found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also, the output comes out as specified requirement by the user. Hence output testing does not result in any correction the system. The project is developed based on the user choice. It is user friendly. The output format is very clear to user. Output testing is done on smart builders correctly.

e) Acceptance testing

Acceptance testing generally involves running a suit of tests on the completed condition of the user's environment or future of the system, and will result in a pass or fail, or Boolean, outcome.

5.SYSTEM IMPLEMENTATION

5.1 SYSTEM DESCRIPTION

The application developed is designed in such a way that any further enhancement can be done with ease. The system has the capability for easy integration. New modules can be added to the system with less effort. The website is developed in python which make it more reliable and compatible with other environments. The application proves better extensibility and flexibility for future enhancements.

5.2 SYSTEM IMPLEMENTATION

The implantation is the final state and it is an important phase. It involves the individual programming, system testing, user training and the operational running of developed proposed system that constitutes the application subsystems. A major task of preparing for implementation is education of users, which should really have been taken place much earlier in the project when they were being involved in the investigation and design work. During the implementation phase system actually takes physical shape. In order develop a system implemented planning is very essential the implementation phase of the software development is concerned with translating design specification into source code. The user tests the developed system and changes are made according to their needs. Our system has been successfully implemented. Before implementation several tests have been conducted to ensure that no errors are encountered during the operation. The implementation phase ends with an evaluation of the system after placing into the operation for a period of time. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from old system to new system. The system can be implemented only after testing is done and is found to be working to specifications. The implementation stage is a systems project in its own right. The implementation stage involves following tasks:

- Careful planning.

- Investigation of system and constraints.
- Design of method to achieve change over
- Evaluation of the changeover method.

In the case of this project all the screens are designed first. For making it to be executable, codes are written on each screen and performs the implementation by creating the database and connecting to the server. After that the system, is Checked, whether it performs all the transactions Correctly. Then databases are cleared and made it to be usable to the technicians.

6.MAINTENANCE

Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes. Maintenance is the ease with which a program can be corrected if any error is encountered, adapted if its environment changes or enhanced if the customer desires a change in requirement. Maintenance follows conversation to extend that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to problems that surface in the system's operation. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

CATEGORIES OF MAINTENANCE

a) Corrective Maintenance

Corrective maintenance is the most commonly used maintenance approach, but it is easy to see its limitations. When equipment fails, it often leads to downtime in production, and sometimes damages other parts. In most cases, this is expensive. Also, if the equipment needs to be replaced, the cost of replacing it alone can be substantial. Reliability of systems maintained by this type of maintenance is unknown and cannot be measured. Corrective maintenance is possible since the consequences of failure or wearing out are not significant and the cost of this maintenance is not great

b) Perfective Maintenance

Modification of a software product after delivery to improve performance or maintainability. This term is used to describe changes undertaken to expand the existing requirements of the system. A successful piece of software tends to be subjected to the Succession of changes resulting in an increase in its requirements. This is based on premise that as the software becomes useful, the user experiment with new cases beyond the of Scope for which it was initially developed. Expansion requirements can take the form enhancement of existing system functionality and improvement in computational efficiency.

c) Adaptive Maintenance

Modification of a software product performed after delivery to keep a are product usable m a changed or changing environment. Adaptive maintenance includes any work initiated as a consequence of moving the software to a different hardware or software platform. It is a change driven by the need to accommodate modifications in the environment of software system. The environment in this context refers to the totality of all conditions and influences which act from outside upon the system. A change to the whole or part of this environment will Warrant a corresponding modification of the software.

d) Preventive Maintenance

Preventive maintenance is a schedule of planned maintenance actions aimed at the prevention of breakdowns and failures. The primary goal of preventive maintenance is to prevent the failure of equipment before it actually occurs. It is designed to preserve and enhance equipment reliability by replacing worn components before they actually fail. Preventive maintenance activities include equipment checks, partial or complete overhauls at specified periods

Long-term benefits of preventive maintenance include:

- ✦ Improved system reliability.
- ✦ Decreased cost of replacement.
- ✦ Decreased system downtime.

7.CONCLUSION

IT Fest Management System is a software developed for fully computerised and online based IT fest management. This system will help in overcoming some drawbacks that were earlier manual system. This software solution help students from various colleges to register through online. This can be used by the organization to manage the events and maintain the records. The IT Fest Management system provide a user-friendly interface easy operation. The software automates the processes that were done manually and that were developed as a partial automated one, thus eliminating the time taken for various activities.

8.BIBLIOGRAPHY

1.<https://www.scribd.com/document/342098952/Synopsis-of-Event-Management-System>

2. <https://solutiondots.com/blog/event-management-system-comprehensive-solutionevents-management.html> 3.

3.http://www.academia.edu/4593395/Project_Report_On_Event_management

.

4.Event Management System by D Yogendra Rao
:<http://www.slideshare.net/DYogendraRao/event-management-system-245928362>.

5.Web based event management system by FakhrunNisha

9.APPENDIX

9.1 SCREEN SHOTS

The screenshot shows the 'College Register Here..' form on the IT Fest Management website. The header includes a 'Demand' icon and a navigation menu with 'Home', 'About Us', 'Registration', and 'Login'. The breadcrumb trail reads 'Home / Student Registration'. The form fields are as follows:

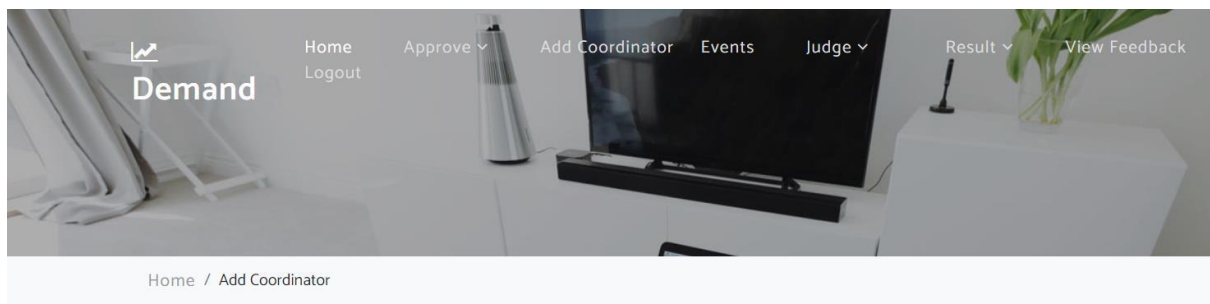
Field	Value
Name	St thomas
Address	St Thomas college Pala
Contact	7025340075
Email	stthomas@gmail.com
Password	*****

A green 'REGISTER' button is located at the bottom of the form.

The screenshot shows the 'Login Here..' form on the IT Fest Management website. The header includes a 'Demand' icon and a navigation menu with 'Home', 'About Us', 'Registration', and 'Login'. The breadcrumb trail reads 'Home / Login'. The form fields are as follows:

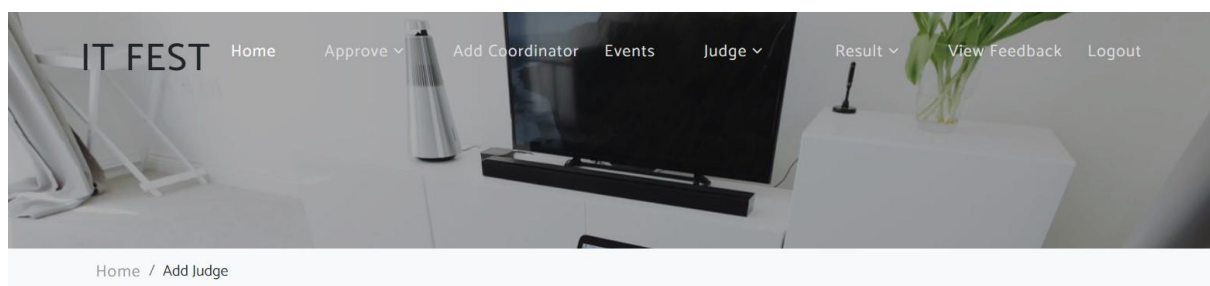
Field	Value
Username	stthomas@gmail.com
Password	*****

A green 'LOGIN' button is located at the bottom of the form.



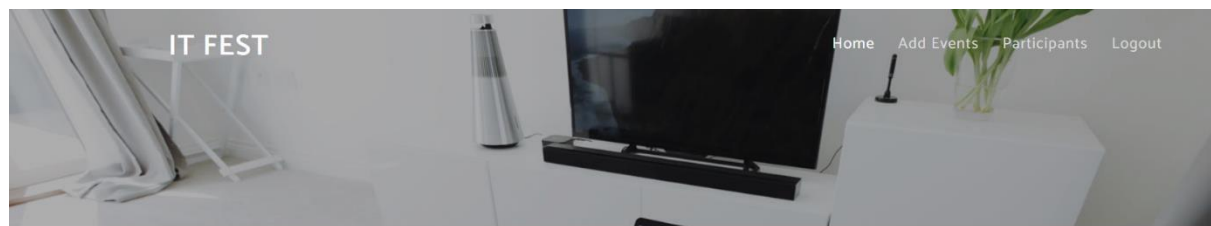
ADD COORDINATOR

Name	<input type="text" value="Sruthy"/>
Address	<input type="text" value="malamuri"/>
Contact	<input type="text" value="9847303717"/>
Place	<input type="text" value="muvattupuzha"/>
Email	<input type="text" value="sruthy@gmail.com"/>
Password	<input type="password" value="*****"/>



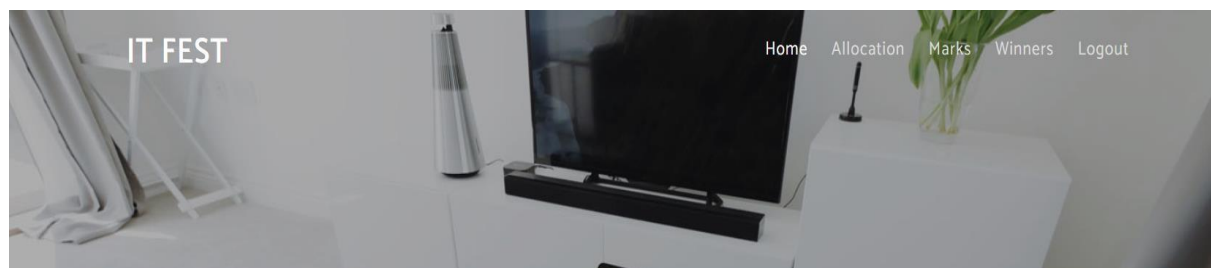
ADD JUDGE

Name	<input type="text" value="Jude thomas"/>
Contact	<input type="text" value="9544842927"/>
Place	<input type="text" value="muvattupuzha"/>
Photo	<input type="button" value="Choose File"/> <input type="text" value="jeeva.jpg"/>
Email	<input type="text" value="jude@gmail.com"/>
Password	<input type="password" value="*****"/>

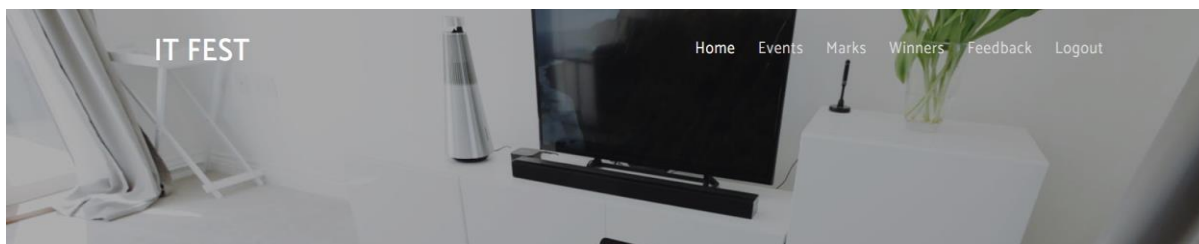


ADD EVENTS

Event Name	<input type="text" value="photography"/>
Date	<input type="text" value="26-06-2021"/>
Venue	<input type="text" value="San Diego"/>
Time	<input type="text" value="11:25"/>
Number Of People	<input type="text" value="10"/>
<input type="button" value="ADD"/>	



Total Score	<input type="text" value="10"/>
<input type="button" value="ADD"/>	



FEEDBACK

Feedback

9.2 CODES

COLLEGE REGISTRATION

```
<div class="contact-top" style="background-color:hsl(0, 0%, 59%);color: white;
width: 500px;margin-left: 350px;border-radius: 30px;">
    <h3 align="center">College Register Here..</h3>
    <br><br>    <form method="POST"
enctype="multipart/form-data">
        {% csrf_token %}
<!-- <h1 align="center">Category</h1> -->
<table align="center">
<tr><th>Name</th><td><input type="text" name="name" required=""
class="formcontrol"></td></tr>
<tr><th>Address</th><td><textarea name="address" required
class="formcontrol"></textarea></td></tr>
<tr><th>Contact</th><td><input type="text" name="contact" maxlength="10"
required="" class="form-control"></td></tr>

<!-- <tr><th> ID Proof</th><td><input type="file" name="idproof" required=""
class="form-control"></td></tr> -->
```

```

<tr><th>Email</th><td><input type="email" name="email" required=""
class="form-control"></td></tr>
<tr><th>Password</th><td><input type="password" name="password" required=""
class="form-control"></td></tr>
<tr><td align="center" colspan="2"><input type="submit" name="submit"
value="REGISTER" class="btn btn-success"></td></tr>
</table> </form>
</div>

```

LOGIN

```

<div class="contact-top">
  <h3 align="center">Login Here..</h3>
  <br><br>    <form method="POST"
enctype="multipart/form-data">
    {% csrf_token %}
<!-- <h1 align="center">Category</h1> -->
<table align="center">

<tr><th>Username</th><td><input type="email" name="username" required=""
class="form-control"></td></tr>
<tr><th>Password</th><td><input type="password" name="password" required=""
class="form-control"></td></tr>
<tr><td colspan="2" align="center"><input type="submit" name="submit"
value="LOGIN" class="btn btn-success"></td></tr>
</table>
</form>
<br><br>
</div>

```

ADD COORDINATOR

```

<div class="contact-top" style="background-color:rgb(161, 161, 161);color:
white; width: 500px;margin-left: 350px;border-radius: 30px;">
    <h3 align="center"></h3>
    <br><br>    <form method="POST"
enctype="multipart/form-data">
        {% csrf_token %}
    <h2 align="center" style="color: blue;font-family: Algerian;">ADD
COORDINATOR</h2> <br>
<table align="center">
<tr><th>Name</th><td><input type="text" name="name" required=""
class="formcontrol"></td></tr>
<tr><th>Address</th><td><textarea name="address" required
class="formcontrol"></textarea></td></tr>
<tr><th>Contact</th><td><input type="text" name="contact" maxlength="10"
required="" class="form-control"></td></tr>
<tr><th>Place</th><td><input type="text" name="place" required=""
class="formcontrol"></td></tr>

<tr><th>Email</th><td><input type="email" name="email" required=""
class="form-control"></td></tr>
<tr><th>Password</th><td><input type="password" name="password" required=""
class="form-control"></td></tr>
<tr><td colspan="2" align="center"><input type="submit" align="center"
name="submit" value="ADD" class="btn btn-success"></td></tr>
</table>
</form>
</div>

```

ADD JUDGE

```

<div class="contact-top" style="background-color:rgb(117, 117, 117);color:
white; width: 500px;margin-left: 350px;border-radius: 30px;">
    <h3 align="center"></h3>
    <br><br>
    <form method="POST" enctype="multipart/form-data">

```

```

        {% csrf_token %}
<h1 align="center" style="color: blue;font-family: Algerian;">ADD
JUDGE</h1><br>
<table align="center">
<tr><th>Name</th><td><input type="text" name="name" required=""
class="formcontrol"></td></tr>

<tr><th>Contact</th><td><input type="text" name="contact" required=""
maxlength="10" class="form-control"></td></tr>
<tr><th>Place</th><td><input type="text" name="place" required=""
class="formcontrol"></td></tr>
<tr><th>Photo</th><td><input type="file" name="image" required=""
class="formcontrol"></td></tr>

<tr><th>Email</th><td><input type="email" name="email" required=""
class="form-control"></td></tr>
<tr><th>Password</th><td><input type="password" name="password" required=""
class="form-control"></td></tr>
<tr><td align="center" colspan="2"><input type="submit" align="center"
name="submit" value="ADD" class="btn btn-success"></td></tr>
</table>
</form>
</div>

```

ADD EVENT

```

<form method="POST" enctype="multipart/form-data">
    {% csrf_token %}
<!-- <h1 align="center">Category</h1> -->
<h3 align="center" style="color: blue;font-family: Algerian;"><b>ADD
EVENTS</b></h3>
<br>
<table align="center">

```



```
<tr><th>Event Name</th><td><input type="text" name="eventname" required=""  
class="form-control"></td></tr>  
  
<tr><th>Date</th><td><input type="date" name="eventdate" required=""  
class="form-control"></td></tr>  
  
<tr><th>Venue</th><td><input type="text" name="place" required=""  
class="formcontrol"></td></tr>  
  
<tr><th>Time</th><td><input type="time" name="time" required=""  
class="formcontrol"></td></tr>  
  
<tr><th>Number Of People</th><td><input type="text" name="nump" required=""  
class="form-control"></td></tr>  
  
<tr><td colspan="2" align="center"><input type="submit" align="center"  
name="submit" value="ADD" class="btn btn-success"></td></tr>  
</table>  
</form>
```

ADD STUDENT

```
<div class="contact-top" style="background-color:rgb(161, 160, 160);color:
white; width: 500px;margin-left: 350px;border-radius: 30px;">
    <h3 align="center">Student Register Here..</h3>
    <br><br>    <form method="POST"
enctype="multipart/form-data">
        {% csrf_token %}
<!-- <h1 align="center">Category</h1> -->
<table align="center">
<tr><th>Name</th><td><input type="text" name="name" required=""
class="formcontrol"></td></tr>
<tr><th>Address</th><td><textarea name="address" required
class="formcontrol"></textarea></td></tr>
<tr><th>Contact</th><td><input type="text" name="contact" maxlength="10"
required="" class="form-control"></td></tr>

<tr><th> ID Proof</th><td><input type="file" name="idproof" required=""
class="form-control"></td></tr>
<tr><th>Email</th><td><input type="email" name="email" required=""
class="form-control"></td></tr>
```

```
<tr><th>Password</th><td><input type="password" name="password" required=""
class="form-control"></td></tr>
<tr><td align="center" colspan="2"><input type="submit" name="submit"
value="REGISTER" class="btn btn-success"></td></tr>
</table>
</form>
</div>
```

ADD MARK

```
<form method="POST" enctype="multipart/form-data">
    {% csrf_token %}
<!-- <h1 align="center">Category</h1> -->
<table align="center">
<tr><th>Total Score</th><td><input type="text" name="tscore"
required=""></td></tr>

<tr><td><input type="submit" align="center" name="submit" value="ADD"
class="btn btn-success"></td></tr>
</table>
</form>
```

ADD FEEDBACK

```
<form method="POST" enctype="multipart/form-data">
    {% csrf_token %}
<!-- <h1 align="center">Category</h1> -->
<table align="center">
<tr><th>Total Score</th><td><input type="text" name="tscore"
required=""></td></tr>
```

```
<tr><td><input type="submit" align="center" name="submit" value="ADD"
class="btn btn-success"></td></tr>
</table>
</form>
```