**SYNOPSIS**

**1. SYNOPSIS**

In today’s competitive environment, getting jobs and searching for candidates assumes greater importance. Earlier, the advertisements of jobs were limited to newspapers. But now the emerging economy demands the situation to change. We have various job portals, online employment exchanges, consultancies, company websites etc. In this situation recruitment should be time saving, cost effective and at the same time should search out the qualified candidates.

By using the efficient methods and techniques the recruitment process is made easy. The employers and candidates see things not in terms of what they need, but in terms of best things they want. The client hunted a job portal which is meant for Employers, jobseekers, and also which guides the candidates in various areas of job search and related things.

Finding and recruiting the best quality candidates seems to get more complicated in the coming days. So the proposed system aimed to provide best class of employment services to job seekers, employers, and recruiters. The system should provide free job search and services. It should be quick, safe and easy to use. The system is meant for the end users -the employer and the candidate. The end users may be registered users or not registered

**INTRODUCTION**

**2. INTRODUCTION**

**2.1 Project Profile**

This project is developed as an online Job Portal for the Placement Dept. of the company. The system is an online application that can be accessed throughout the organization and outside as well with proper login provided. This system can be used as an Online Job Portal for the Placement Dept. of the organization to manage the student information with regards to placement. Students logging should be able to upload their information in the form of a CV. Visitors/Company representatives logging in may also access/search any information put up by Students.

The project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. Using the constructs of MS-SQL Server and all the user interfaces have been designed using theASP.Net technologies. The database connectivity is planned using the “SQL Connection” methodology. The standards of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports, which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.

The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The specification has been normalized up to 3NF to eliminate all the anomalies that may arise due to the database transaction that are executed by the general users and the organizational administration. The user interfaces are browser specific to give distributed accessibility for the overall system. The internal database has been selected as MS-SQL server 200.The basic constructs of table spaces, clusters and indexes have been exploited to provide higher consistency and reliability for the data storage. The MS-SQL server 200 was a choice as it provides the constructs of high-level reliability and security. The total front end was dominated using the ASP.Net technologies. At all proper levels high care was taken to check that the system manages the data consistency with proper business rules or validations.

**ABOUT THE DEVELOPING TOOL**

**3. ABOUT THE DEVELOPING TOOLS**

**3.1 Introduction to PHP**

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by [Rasmus Lerdorf](https://en.wikipedia.org/wiki/Rasmus_Lerdorf) in 1994,  the PHP reference implementation is now produced by The PHP Development Team.PHP originally stood for Personal Home Page,but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML or HTML5 code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server. PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.

**3.2 Features of PHP**

1. In PHP there is no need to specify a data type for variable declaration. Rather, it can be determined at the time of execution depends on the value of the variable. So that, PHP is called as loosely typed language.
2. PHP provides cross-platform compatibility, unlike some other server-side scripting language.
3. PHP has set of predefined variables called super globals which will start by \_. Some of the examples are, $\_GET, $\_POST, $\_COOKIE, $\_SESSION, $\_SERVER and etc. So, any variable except super globals, that start with \_ will cause an error.
4. PHP programming structure includes variable variables; that is, the name of the variable can be changed dynamically.
5. This language contains access monitoring capability to create logs as the summary of recent accesses.
6. And then, it includes several magic methods that begin with \_\_ character which will be defined and called at appropriate instance. For example, \_\_clone() will be called, when the clone keyword is used.
7. Predefined error reporting constants are available to generate a warning or error notice. For example, when E\_STRICT is enabled, a warning about deprecated methods will be generated.
8. PHP supports extended regular expression that leads extensive pattern matching with remarkable speed.
9. And then, properties like, nowdocs and heredocs are used to delimit some block of context which should not be sent for parsing.
10. Since PHP is a single inheritance language, the parent class methods can be derived from only one directly inherited subclass. But, the implementation of traits concept, reduce the gap over this limitation and allow to reuse required method in several classes.

**3.3 CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and

formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, andfonts.] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable.

**3.4 HTML**

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <IMG/>and <INPUT/> introduce content into the page directly. Others such as <P>…</P> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

**3.5 MySQL**

MySQL is the world's most widely used open-source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases, though SQLite probably has more total embedded deployments. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web

Application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL. MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

**3.6 Operating System-LINUX**

Linux is one of popular version of UNIX operating System. It is open source as its source code is freely available. It is free to use. Linux was designed considering UNIX compatibility. Its functionality list is quite similar to that of UNIX.

The OS is comprised of a number of pieces:

* **The Bootloader:** The software that manages the boot process of your computer. For most users, this will simply be a splash screen that pops up and eventually goes away to boot into the operating system.
* **The kernel:** This is the one piece of the whole that is actually called “Linux”. The kernel is the core of the system and manages the CPU, memory, and peripheral devices. The kernel is the “lowest” level of the OS.
* **Daemons:** These are background services (printing, sound, scheduling, etc) that either start up during boot, or after you log into the desktop.
* **The Shell:** You’ve probably heard mention of the Linux command line. This is the shell – a command process that allows you to control the computer via commands typed into a text interface. This is what, at one time, scared people away from Linux the most (assuming they had to learn a seemingly archaic command line structure to make Linux work). This is no longer the case. With modern desktop Linux, there is no need to ever touch the command line.
* **Graphical Server:** This is the sub-system that displays the graphics on your monitor. It is commonly referred to as the X server or just “X”.
* **Desktop Environment:** This is the piece of the puzzle that the users actually interact with. There are many desktop environments to choose from (Unity, GNOME, Cinnamon, Enlightenment, KDE, XFCE, etc). Each desktop environment includes built-in applications (such as file managers, configuration tools, web browsers, games, etc).
* **Applications:** Desktop environments do not offer the full array of apps. Just like Windows and Mac, Linux offers thousands upon thousands of high-quality software titles that can be easily found and installed. Most modern Linux distributions (more on this in a moment) include App Store-like tools that centralize and simplify application installation. For example: Ubuntu Linux has the Ubuntu Software Center (Figure 1) which allows you to quickly search among the thousands of apps and install them from one centralized location.

**Basic Features**

Following are some of the important features of Linux Operating System.

* **Portable** − Portability means software can works on different types of hardware in same way. Linux kernel and application programs supports their installation on any kind of hardware platform.
* **Open Source** − Linux source code is freely available and it is community based development project. Multiple teams work in collaboration to enhance the capability of Linux operating system and it is continuously evolving.
* **Multi-User** − Linux is a multiuser system means multiple users can access system resources like memory/ ram/ application programs at same time.
* **Multiprogramming** − Linux is a multiprogramming system means multiple applications can run at same time.
* **Hierarchical File System** − Linux provides a standard file structure in which system files/ user files are arranged.
* **Shell** − Linux provides a special interpreter program which can be used to execute commands of the operating system. It can be used to do various types of operations, call application programs. etc.
* **Security** − Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

**SYSTEM ANALYSIS**

**4. SYSTEM ANALYSIS**

**4.1 Introduction**

System analysis refers to an orderly structured process for identifying and solving problems using computer. It is the most essential part of the project development. System analysis is a process of gathering and interpreting facts diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the user’s and system developers. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action. To analyze a system one has to study the various aspects of how the current system works and what are the needs of the user. In other words system analysis specifies what the system should do.

The analysis is intended to capture and describe all the requirements of the system and to make a model that defines the key domain classes of the system. The purpose is to provide an understanding and enable a communication about the system between the developers and the people establishing the requirements. Therefore the analysis is typically conducted in cooperation with the user or customer.

* 1. **Existing System**

The existing system is common to all kind of job seekers. All kind of jobs will be shown without considering the applicants qualification to the job which makes the job seeker very difficult to find the offers in which they are qualified to. Job providers also can post their job without any approval of admin, hence there are possibilities of fake offers.

**Functional Components of the project:**

* To facilitate easy maintenance of records of various Recruiters (Companies), job and job seekers.
* To check for details prospective jobseekers through quick search provided in the portal.
* To check for matching job with jobseekers.
* Quick access of all record.
* To match the suitable candidates to appropriate job.
* Prevent and reduce human error.
* Reduce manual work.
  + 1. **The Limitations of existing system**
* The limitations of existing job portal system are that according to users qualification sorting their preferred jobs is not possible.
* Complicated system for students and fresher to find their respective jobs.
* Fake offers can’t be avoided because the approval of admin is not required.
  1. **Feasibility Study**

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

 Economic Feasibility

 Technical Feasibility

 Operational Feasibility

**4.3.1 Economic feasibility**

This involves questions such as whether the firm cans effort to build the system, whether the benefits should substantially exceed its costs and whether the project has higher priority and profits than other projects that might use same resources. This also includes that whether the project is in the condition to fulfill all the eligibility criteria and responsibility of both sides in case there are two parties in performing any project.

**4.3.2 Technical feasibility**

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build and whether the firm has enough experience using that technology. The assessment is based on an outline design of system requirements in terms of data, trends, frequency of updating, etc…in order to give an introduction of technical system.

**4.3.3 Operational feasibility**

The analyst considers the extent the proposed system will fulfill his departments. That is whether the proposed system covers all aspects of the working system and whether it has considerable improvements.

* 1. **Proposed System**

The new system that was to be built in the organization need to have some creative concept that can help the user in the real manner and the next important thing is that it should give the cost effective solution to the user. Due to the collaborative nature of the application the user can really be an important part of it rather than just using it blindly. The new application also gives the user the liberty to know and use the application from the web through the web modules which gives complete information of the application.

* + 1. **Advantages of proposed system**

The key features of online job portal are:

* Preferably to be used for students and fresher which takes care of the complexity.
* Only the suitable jobs will be shown to the user according to their qualification.
* Job providers can only post the job by the approval of admin; hence fake offers can be avoided.

**SYSTEM SPECIFICATION**

**5. SYSTEM SPECIFICATION**

* 1. **Software Specification**
* Operating System : Linux
* Front End : PHP, CSS, HTML
* Back End : MySQL
* Browser : Google Chrome
  1. **Hardware Specification**
* Processor : Pentium (Iv) Or Higher
* Memory : 2 GB or above
* Mother Board : Intel 815e
* Ram : 256 MB
* Hard Drive: 50 GB
* Key Board : 104 Keys Standard Key Board
* Monitor : Crt/Lcd
* Mouse : Scroll Mouse

**SYSTEM DESIGN**

**6. SYSTEM DESIGN**

**6.1 Introduction to System Design**

Design is the phase that indicates the final system. It is the solution, the translation of requirements into ways of meeting them. In this phase the following elements were designed namely, dataflow, data stores, processes, procedures. Firstly the logical design was done where the output, inputs and databases and procedures was formulated in a manner that meets the project requirements. After logical design physical construction of the system is done. After analyzing the various functions involved in the system the database and tables are designed. Care is taken for the field name to be in self-explanatory form. Unnecessary fields are avoided so as not to affect the storage of the system. Care must be taken to design the input screen in the most user friendly way so as to help even the novice users make entries approximately in the right place.

All input screens in the system are user friendly. The sizes of all the screens are standardized. The importance of the software design can be stated with a single word quality. Design is a place where quality is fostered in software development. Design is the only way where requirements are actually translated into a finished software product or system**.** In this project tables are create in MySQL.

The first step in the design phase is to design the database and then input and output within predefined guidelines. Simple designs are easily understood, easily built, and easily tested. Simplicity is the most important criteria of a design. Other design criteria include the following.

**Documentation:** A good design always comes with a set of well written documents.

**Testability:** In a good design every requirement is testable. A design that cannot be easily tested against its requirements is not acceptable design.

**Structure:** A good design presents hierarchical structure that makes logical use of control policies among components.

**Modularity:** A good design is modular and exhibits the properties of high cohesiveness and low coupling.

**Discreteness:** A good design separates data procedures and timing consideration to the extent possible.

**Representation:** A good design should be easily communicated to all interested parties through appropriate abstraction and representation.

**Reusability:** A good design should be repeatable and reusable.

The major step in design is the preparation of input and the design of output reports in a form which is acceptable to the user.

* 1. **Input Design**

Inaccurate input data are most common cause of errors in data processing. Errors entered by data entry operators can be controlled by input design. Input design is the process of converting user-oriented inputs to a computer-based format. Input data are collected and organized into groups of similar data.

The goal of designing input data is to make data entry easy, logical and free from errors as possible. In the design of input the following steps must be considered.

 The allocated space for each field.

 Field sequence, which must match that in the source document.

 The format in which data fields are entered

We have to keep in mind the following things to design the system

 What data to input

 What medium to use

 The dialogue to guide users in providing input

 Methods for performing input validation and steps to follow when errors occur.

Input design is a part of overall system design which requires very careful attention. Often the collection of input data is the most expensive part of the system, in terms of the equipment used; it is the point of most contact for the users with the computer system; and it is prone to error. If data going into the system is incorrect, then the processing and output will magnify these errors. Thus the designer has a number of clear objectives in input design.

In this project there are several forms to get the inputs such as email id and username to register the reporter as well as admin. After the registration the reporter can add the news, jobs and event details.

**6.3 Output Design**

Outputs from computer systems are required primarily to communicate the result of processing to users or sometimes to other systems, including machine – based systems. They are also used to provide a permanent copy of these results for later consultation. These are various types of output required by most systems, the main ones are:

 **External Outputs:**

Whose destination is outside the organization and which require special attention because they project the image of the organization.

 **Internal Outputs:**

Whose destination is within the organization and which require careful design because they are the users’ among interface with the computer.

 **Operational Outputs:** Whose use is purely within the computer department, e.g. program listings, usage statistics etc.

 **Interactive Outputs:**

This involves the user in communicating directly with the computer.

 **Output Definition:**

The outputs should be defined in terms – type of output, content, format, location, sequence.

The proposed output of this project is an android mobile news application which includes latest job, news and event details.

* 1. **Form Design**

The forms are used to show the different pages that we use to implement in the proposed system. They can be used to implement different styles and features for the respective pages.

The interface design describes how the software communicates within itself, with system that interoperates with it, and with humans who use it. An interface implies a flow of information and a specific type of behavior. Therefore, data and control flow diagrams provide much of the information required for interface design.

The forms or the interface or entering the user and administrator information have been designed with labels that suggest about the right input and validation controls which prevents wrong data to be entered through these forms to enter the database. The forms are designed in such a way that the user will find it very easy and convenient to use these forms efficiently.

The main forms used in this project are Admin registration and User registration. Admin and User can register through these forms.

* 1. **System Modules**

The system contains the following modules.

* **Admin module:** The system administrator can monitor the system, The Admin can add or approve the users and company.
* **User module**: The user can manage their accounts like adding or updating their profile details. The user can also search and apply for their required job.
* **Company module:** The company can post their job vacancies and also they can update their profile details too.
  1. **Data Flow Diagram**
     1. **Introduction to DFD**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel unlike a flowchart which also shows this information.

DFD Notations

It represents a process or transform that is applied to data

It represents data store-stored information that is used by software

It represents a source or destination

It represent a flow of data,that is,a data stream

* + 1. **Proposed System DFD**

**Level 0**

****

**Level 1**

****

**Level 2**

****

****

****

****

* 1. **Table Structure**

1. Table name: **admin**

Description: Store details of admin

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| id\_admin | Integer (5) | Primary Key  Auto increment | Admin id |
| username | Varchar(50) | Not Null | Username of admin |
| password | Varchar(50) | Not Null | Password of the admin |

1. Table name: **users**

Description: Store registration details of the user

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| userid | Integer (5) | Primary Key  Auto increment | User id |
| fname | Varchar(50) | Not Null | First name of user |
| lname | Varchar(50) | Not Null | Last name of user |
| email | Varchar(50) | Not Null | Email Id of user |
| password | Varchar(50) | Not Null | Password of user |
| address | Text |  | Address of user |
| city | Varchar(50) |  | City of user |
| state | Varchar(50) |  | State of user |
| country | Varchar(50) |  | Country of user |
| contactno | Varchar(50) |  | Contact of user |
| qualification | Varchar(50) |  | Qualification of user |
| stream | Varchar(50) |  | Stream of course |
| passingyear | Varchar(50) |  | Year of passing |
| dob | Varchar(50) |  | Date of Birth of user |
| age | Varchar(50) |  | Age of user |
| designation | Varchar(50) |  | Designation of user |
| photo | Varchar(255) |  | Profile picture of user |
| cv | Varchar(255) |  | Resume of user |
| active | Integer (11) | Not Null | Approval Status |

1. Table name: **company**

Description: Store registration details of company

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| id\_company | Integer(5) | Primary Key  Auto increment | Company id |
| companyname | Varchar(50) | Not Null | Company name |
| contactno | Varchar(50) | Not Null | Contact of company |
| website | Varchar(50) |  | Website of company |
| email | Varchar(50) | Not Null | Email of company |
| password | Varchar(50) | Not Null | Password of company |
| country | Varchar(50) |  | Country of company |
| state | Varchar(50) |  | State of company |
| city | Varchar(50) |  | City of company |
| aboutme | Varchar(50) |  | About the company |
| logo | Varchar(255) |  | Logo of company |
| createdAt | Timestamp | Not Null | Creation Time |
| active | Integer (11) | Not Null | Approval Status |

1. Table name: **userqualification**

Description: Store qualification details of users

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| q\_id | Integer (5) | Primary Key  Auto increment | Qualification Id |
| user\_id | Varchar(50) | Not Null | User Id |
| q\_level | Varchar(50) | Not Null | Qualification Level |
| qualification | Varchar(50) | Not Null | Qualification |
| subject | Varchar(50) | Not Null | Specialized Subject |
| institution | Varchar(50) |  | Institution Name |
| university | Varchar(50) |  | University Name |
| percentage | float |  | Percentage mark |
| grade | Varchar(2) |  | Grade obtained |
| passout | Integer (11) |  | Pass out year |
| register\_number | Integer (100) |  | Register Number |

1. Table name: **job\_post**

Description: Store Job posts details

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| id\_jobpost | Integer(5) | Primary Key  Auto increment | Jobpost id |
| id\_company | Integer(5) | Not Null | Company id |
| jobtitle | Varchar(50) | Not Null | Job Title |
| description | Varchar(500) | Not Null | Job Description |
| minimumsalary | Varchar(50) | Not Null | Minimum Salary |
| maximumsalary | Varchar(50) | Not Null | Maximum Salary |
| experience | Varchar(50) | Not Null | Experience required |
| ug\_course | Varchar(50) |  | UG Course |
| ug\_mark | Integer(5) |  | UG Percentage |
| pg\_course | Varchar(50) |  | PG Course |
| pg\_mark | Integer(5) |  | PG Percentage |
| createdAt | timestamp | Not Null | Jobpost created time |
| active | Integer(11) | Not Null | Approval Status |

1. Table name: **apply\_job**

Description: Store details of users applied for jobs

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| id\_applyjob | Integer(5) | Primary Key  Auto increment | Jobpost id |
| id\_company | Integer(5) | Not Null | Company id |
| id\_jobpost | Integer(5) | Not Null | Jobpost id |
| id\_user | Integer(5) | Not Null | User id |

1. Table name: **admin\_mailbox**

Description: Store mails of admin

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| id\_mail | Integer(5) | Primary Key  Auto increment | Mail id |
| id\_company | Integer(5) |  | Company id |
| id\_user | Integer(5) |  | User id |
| mail\_title | Varchar(50) | Not Null | Mail id |
| mail\_content | Varchar(255) | Not Null | Mail Content |
| createdAt | timestamp | Not Null |  |

**SYSTEM TESTING**

**7. SYSTEM TESTING**

* 1. **Introduction to System Testing**

Software testing is critical element of software quality assurance and represents the ultimate review of specification, design and coding. Software testing is one broader topic and often referred to verification and validation. These aspects refer to all activities that ensure that the software performs a specific function and different set of activities that ensure that the software built is traceable to user requirements. The testing procedure mainly consists of the following:

 Unit Testing

 Integration Testing

 Validation Testing

**7.2 Unit Testing**

This is the first level of testing. A Unit Testing focuses verification effort on the smallest unit of software design. This testing was carried out during the coding itself. In this testing step, each module is going to satisfactorily as expected output from the module. After coding each dialogue has tested and run individually. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence the naming is unit testing.

Each smaller unit starting from the registration of admin and reporter is tested. After that the following modules news management, event management and job management is also tested.

**7.3 Integration Testing**

Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing. That is, integration testing is the complete testing of the set of modules, which makes up the products. The objectives are to take untested modules and build a program structure that has been dictated by design as integration testing is conducted, the tester should identify the critical modules.

One approach is to wait all the units have passed the testing, and then combine them and then tested. Another strategy is to construct the product is increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination, and so on. The advantage of this approach is that, interface dispenses can be easily found and corrected. Testing is completed when the last module is integrated and testing.

Integration testing done after the development of the mobile application by combining all the tests.

**7.4 Validation Testing**

At the culmination of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected, and validation testing may begin. Validation succeeds when the software function in a, manner that can be reasonably expected by customer.

Software validation is achieved through a series of black box tests that demonstrates conformity with requirements. A test plan outlines the classes of tests to be conducted and a test procedure defines specific test cases that will be used to demonstrate conformity with requirements. Both the plan and procedure designed to ensure that all functional requirements are achieved, documentation is correct and human-engineered, and other requirements are met. Validation testing was done by inputting dummy data to ensure that software developed is satisfied all the requirements of the user. A test case is a set of data that the system will process as normal input. However, the data are created with the intent of determining whether the system will process them correctly.

Validation testing is done for each modules. During the registration process email is validated and number of job vacancies are validated on job management module.Time and date is also validated.

**7.5 Test Cases**

Project Title : Online job portal

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case No | Test Data | DB Table  Name(s)  influenced | Form(s)/  Report(s)  Involved | Expected Result | Actual Result | Remarks |
| 1. | uname:admin  password:admin | Login | Admin.php | Admin Home Page | Logged Successfully | Passed |
| 2. | uname:admin Password: 1234 | Login | Admin.php | Admin Home Page | Invalid user | Failed Incorrect password |
| 3. | uname:raju@gmail.com  password:raj | Login | Login.php | User Home Page | Logged Successfully | Passed |
| 4. | uname:raju@gmail.com Password: raju123 | Login | Login.php | User Home Page | Invalid user | Failed Incorrect password |
|  | Register New User | User | Adduser.php | Registered Successfully | Registered Successfully | Passed |
|  | Register New Company | Company | addcompany.php | Created Successfully | Created Successfully | Passed |
|  | Create New Job Post | Jobs | addjobs.php | Created Successfully | Created Successfully | Passed |
|  | Apply job | Jobs | applyjob.php | Created Successfully | Created Successfully | Passed |
| 9. | Create mail | Mail | Addmail.php | Created Successfully | Created Successfully | Passed |
| 10. | Create mail | Mail | Addmail.php | Created Successfully | Created Successfully | Passed |

**SYSTEM IMPLEMENTATION**

**8. SYSTEM IMPLEMENTATION**

**8.1 Introduction to System Implementation**

Implementation is the stage of the project where theoretical design is turned into a working system. If the implementation is not carefully planned and controlled, it can cause chaos and confusion. Proper implementation is essential to provide a reliable system to meet organization requirements. Successful implementation may not guarantee improvement in the organization using the system, but proper installation will prevent it. The process of putting the developed system in actual use is called system implementation. The system can only be implemented after thorough testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system.

The implementation is the final stage and it is an important phase. It involves individual programming, system testing and operational running of the developed system that constitute the application subsystem. During the implementation phase the system actually takes the physical shape. In order for the system implementation planning is necessary. The implementation phase of the software is concerned while design specification into source code. The user tests the developed system and changes are made according to suite his/her needs. Our system has been successfully implemented before several tests have been conducted to ensure that no errors are encountered during the operation, in case of errors they have to be rectified effectively. Errors can be of various types, mainly minor and major, requiring the corresponding effort. Even a dot or comma may sometimes cause major errors.

The implementation phase ends with an evaluation of the system after placing it into operation for a period of time. In order to achieve the objectives and the expected performance, the system has been developed in a highly interactive and user-friendly manner. The system implementation was carried out using five main aspects:

* Transition planning.
* Training.
* Security.
* Protection.
* Quality control.
* Installation Procedure

In the field of computer software, the term software build refers either to the process of converting source code file into standalone software artifacts that can be run on a computer, or the result of doing so. One of the most important steps of a software building is the compilation process where source code files are converted into executable code.

1. We have to make sure that all applications have to be closed before the installation.

2. Install the software in all the client system in the network.

For implementing the project, the different software must be installed for its fast and better execution. To install the system, the primary need is web based environments without which the system will not have a proper utilization.

**8.2 Training**

An Implementation plan is a management tool for a specific policy measure, or package of measures, designed to assist agencies to manage and monitor implementation effectively. Implementation plans are intended to be scalable and flexible; reflecting the degree of urgency, innovation, complexity and/or sensitivity associated with the particular policy measure. Agencies are expected to exercise judgment in this area; however, the level of detail should be sufficient to enable the agency to effectively manage the implementation of a policy measure.

* 1. **Conversion**

Whether establishing a plan for the first time or converting from an existing one, we understand that ensuring a well-designed process is critical for success. With a history of implementing thousands of plans per year, we’ve found that two things are necessary for success: a smooth plan installation and effective communication with employees.

* 1. **Post Implementation And Review**

A Post-Implementation Review (PIR) is an assessment and review of the completed working solution. It will be performed after a period of live running, sometime after the project is completed. There are three purposes for a Post-Implementation Review:

 To ascertain the degree of success from the project, in particular, the extent to which it met its objectives, delivered planned levels of benefit, and addressed the specific requirements as originally defined.

 To learn lessons from this project, lessons which can be used by the team members and by the organization to improve future project work and solutions.

 To examine the efficacy of all elements of the working business solution to see if further improvements can be made to optimize the benefit delivered.

 In some cases, the first of these objectives can be a contractual issue. Where that is the case, it may be safer to run separate reviews - one focused on contractual compliance and the other seeking to derive further benefit from a no-blame review.

**8.5 System Maintenance**

Once the software is fully developed and implemented, the company starts to use the software. The company also grows and more divisions can be attached to the company, or the database of the company can grow in size. So after some time the software, which has been installed, needs some modification. If the software needs modification all the steps needed to develop new software has to be executed. Maintenance can be classified as corrective, adaptive or perceptive.

* **Corrective maintenance:** Reactive modification of a software product performed after delivery to correct discovered problems.
* **Adaptive maintenance:** Modification of a software product performed after delivery to keep a software product usable in a changed or changing environment.
* **Perfective maintenance:** Modification of a software product after delivery to improve performance or maintainability.

**CONCLUSION**

**9. CONCLUSION**

The project entitled “Online job portal” was successfully designed, developed and tested. The given objectives are met with satisfaction. The Online job portal aims at developing an website that enables the user to deal with recent jobs and attempt to minimize the problems of an applicant to find a correct job.

An administrator who manages all the internal activities. The activities include validating the users like jobseekers and job providers, remove fake offers. The website is developed in PHP using SQL server. All these can provide better services to the users of this system.

This is to conclude that the project that we undertook was worked upon with a sincere effort. Most of the requirements have been fulfilled up to the mark and the requirements which have been remaining, can be completed with a short extension.

**APPENDIX**

**10. APPENDIX**

**10.1 APPENDIX A**

**Sample source code/Pseudo code**

**index.php**

<?php

session\_start();

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Onlooker News- Login</title>

<link rel="stylesheet" href="css/style1.css" media="screen" type="text/css" />

<link rel="stylesheet" href="http://fonts.googleapis.com/css?family=Open+Sans:400,700">

<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>

<style>

.btnDown

{

background-color:green;

color:white;

width:300px;

height:80px;

}

</style>

</head>

<body>

<div class="container">

<div id="login">

<form action="" method="POST">

<fieldset class="clearfix">

<center><img src="images/logo.png" width=200px height=200px /></center>

<p><span class="fontawesome-user"></span><input type="text" name="txtuname" value="Username" onBlur="if(this.value == '') this.value = 'Username'" onFocus="if(this.value == 'Username') this.value = ''" required></p> <!-- JS because of IE support; better: placeholder="Username" -->

<p><span class="fontawesome-lock"></span><input type="password" name="txtpass" value="Password" onBlur="if(this.value == '') this.value = 'Password'" onFocus="if(this.value == 'Password') this.value = ''" required></p> <!-- JS because of IE support; better: placeholder="Password" -->

<p><input type="submit" value="Log In" name="txtsubmit"></p>

</fieldset>

</form>

<br><br>

</div> <!-- end login -->

</div>

</body>

</html>

<?php

$f=1;

if(isset($\_POST['txtsubmit']))

{

$conn = mysqli\_connect("localhost","u558220466\_user1","onlookernews123")or die(mysqli\_error());

mysqli\_select\_db($conn,"u558220466\_db1");

$r1=mysqli\_query($conn,"select LoginId,Email,Password from Login");

while($v=mysqli\_fetch\_array($r1,MYSQL\_ASSOC))

{

$uname=$v['Email'];

$pass=$v['Password'];

$logid=$v['LoginId'];

if($uname==$\_POST['txtuname'] && $pass==$\_POST['txtpass'] && $logid=="1")

{

$f=1;

$\_SESSION['username']=$uname;

header("Location:AdminHome.php");

}

else if($uname==$\_POST['txtuname'] && $pass==$\_POST['txtpass'])

{

$r2=mysqli\_query($conn,"select ReporterId,Email from Reporter");

while($v1=mysqli\_fetch\_array($r2,MYSQL\_ASSOC))

{

$rpid=$v1['ReporterId'];

$email=$v1['Email'];

if($email==$uname)

{

$\_SESSION['reporterid']=$rpid;

$\_SESSION['email']=$email;

}

}

$f=1;

$\_SESSION['username']=$uname;

header("Location:ReporterHome.php");

}

else

{

$f=0;

}

}

if($f==0)

{

echo '<script language="javascript">alert("Invalid username or password!!!")</script>';

}

}

?>

**NewNews.php**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="description" content="Miminium Admin Template v.1">

<meta name="author" content="Isna Nur Azis">

<meta name="keyword" content="">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Onlooker News - New News</title>

<!-- start: Css -->

<link rel="stylesheet" type="text/css" href="asset/css/bootstrap.min.css">

<link href="asset/css/style.css" rel="stylesheet">

<!-- end: Css -->

<link rel="shortcut icon" href="asset/img/logomi.png">

<style>

.menuBtn

{

width:210px;

height:150px;

background-color: #A90348;

color: #ffffff;

border-color: #4CAF50;

font-size: 17px;

border: 1px solid #cccccc;

border: none;

display: inline-block;

outline: 0;

padding: 50px;

margin:20px;

vertical-align: middle;

overflow: hidden;

text-decoration: none !important;

text-align: center;

cursor: pointer;

white-space: nowrap;

}

.pubBtn

{

width:100px;

height:30px;

background-color: #4CAF50;

color: #ffffff;

border-radius:25px;

}

.rejBtn

{

width:100px;

height:30px;

background-color: #ff0000;

color: #ffffff;

border-radius:25px

}

.newsDiv{

width:100%;

padding:5px;

}

.newsimgDiv{

float:left;

width:15%;

height:300px;

}

.newtitleDiv{

float:left;

width:85%;

height:300px;

overflow-y: auto;

padding:10px;

}

</style>

</head>

<body id="mimin" class="dashboard">

<?php

session\_start();

echo $\_SESSION['username'];

if($\_SESSION['username']=="")

header("Location:index.php");

?>

<!-- start: Header -->

<nav class="navbar navbar-default header navbar-fixed-top">

<div class="col-md-12 nav-wrapper">

<div class="navbar-header" style="width:100%;">

<a href="ReporterHome.php" class="navbar-brand">

<img src="images/logoLong.png" width="200px" height="30px" /> </a>

<ul class="nav navbar-nav search-nav">

<li><div class="search"></div> </li> </ul>

<ul class="nav navbar-nav navbar-right user-nav">

<li class="dropdown avatar-dropdown">

<img src="asset/img/avatar.jpg" class="img-circle avatar" alt="user name" data-toggle="dropdown" aria-haspopup="true" aria-expanded="true"/>

<ul class="dropdown-menu user-dropdown">

<li><a href="#"><span class="fa fa-user"></span> </a></li>

<li><a href="#"><span class="fa fa-calendar"></span> </a></li>

<li role="separator" class="divider"></li>

<li class="more">

<ul>

<!--

<li><a href=""><span class="fa fa-cogs"></span></a></li>

<li><a href=""><span class="fa fa-lock"></span></a></li>

-->

<li><a href=""><span class="fa fa-power-off "><a href="">&nbsp; Logout</a></span></a></li>

</ul>

</li>

</ul>

</li>

<li class="user-name"><span><?php echo $\_SESSION['username'];

?>

</span></li>

<li><form method="POST" action=""> <input type="submit" name="btnlogout" value="Logout" style="margin:10px;width:70px;height:35px;background-color:red;" /></form></li></ul></div></div></nav>

<!-- end: Header -->

<div class="container-fluid mimin-wrapper">

<!-- start: content -->

<div id="content">

<div style="width:100%;">

<?php

session\_start();

$conn = mysqli\_connect("localhost","u558220466\_user1","onlookernews123")or die(mysqli\_error());

mysqli\_select\_db($conn,"u558220466\_db1");

$sql = "SELECT NewsId FROM News order by NewsId desc limit 1";

$result = $conn->query($sql);

if ($result->num\_rows > 0)

{

while($row = $result->fetch\_assoc()) {

$NewsId=$row["NewsId"]+1;}

}

else

{

$NewsId=1;

}

?>

<div class="form-style-5">

<form action="" method="post" enctype="multipart/form-data" name="form1">

<fieldset>

<legend> Create New News</legend>News Id

<input type="text" name="newsid" value='<?php echo $NewsId; ?>' disabled><font color=red>\*</font>

News Title

<input type="text" name="title" required><font color=red>\*</font>

Description

<textarea name="add" rows="5" cols="20" required></textarea><font color=red>\*</font>

Date

<input type="date" name="date" required><font color=red>\*</font>

Time

<input type="time" name="time" required>

Image<br>

<input type="file" name="fileToUpload" id="fileToUpload">

<br><br>

<input type="submit" name="btn" value="Save" >

</fieldset>

</form>

</div>

<style type="text/css">

.form-style-5{

max-width: 500px;

padding: 10px 20px;

background: #f4f7f8;

margin: 10px auto;

padding: 20px;

background: #f4f7f8;

border-radius: 8px;

font-family: Georgia, "Times New Roman", Times, serif;

}

.form-style-5 fieldset{

border: none;

}

.form-style-5 legend {

font-size: 1.4em;

margin-bottom: 10px;

}

.form-style-5 label {

display: block;

margin-bottom: 8px;

}

.form-style-5 select{

-webkit-appearance: menulist-button;

height:35px;

}

.form-style-5 input[type="submit"]:hover,

.form-style-5 input[type="button"]:hover

{

background: #109177;

}

</style>

</div> </div>

<!-- end: content -->

</div>

</body>

</html>

<?php

if(isset($\_POST["btn"]))

{

$title=$\_POST['title'];

$description=$\_POST['add'];

$time=$\_POST['time'];

$date=$\_POST['date'];

$status="Created";

$file1=$\_FILES["fileToUpload"]["name"];

$rpid=$\_SESSION["reporterid"];

//for upload image

$target\_dir = "uploads/";

$target\_file = $target\_dir . basename($\_FILES["fileToUpload"]["name"]);

$uploadOk = 1;

$imageFileType = pathinfo($target\_file,PATHINFO\_EXTENSION);

$conn = mysqli\_connect("localhost","u558220466\_user1","onlookernews123")or die(mysqli\_error());

mysqli\_select\_db($conn,"u558220466\_db1");

if(empty($file1))

{

mysqli\_query($conn,"insert into News values(DEFAULT,'$title','$description','$time','$date','','$status',$rpid)");

echo '<script language="javascript">alert("Saved successfully")</script>';

header("Location:ReporterHome.php");

exit();

die();

}

$check = getimagesize($\_FILES["fileToUpload"]["tmp\_name"]);

if($check !== false) {

echo "File is an image - " . $check["mime"] . ".";

$uploadOk = 1;

} else {

echo "File is not an image.";

$uploadOk = 0;

}

// Check if file already exists

if (file\_exists($target\_file)) {

//echo "Sorry, file already exists.";

$uploadOk = 0;

}

// Check file size

if ($\_FILES["fileToUpload"]["size"] > 500000) {

echo "Sorry, your file is too large.";

$uploadOk = 0;

}

// Allow certain file formats

if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"

&& $imageFileType != "gif" ) {

echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";

$uploadOk = 0;

}

// Check if $uploadOk is set to 0 by an error

if ($uploadOk == 0) {

echo "Sorry, your file was not uploaded.";

// if everything is ok, try to upload file

} else {

if (move\_uploaded\_file($\_FILES["fileToUpload"]["tmp\_name"], $target\_file)) {

//echo "The file ". basename( $\_FILES["fileToUpload"]["name"]). " has been uploaded.";

} else {

echo "Sorry, there was an error uploading your file.";

}

}

$pic=basename( $\_FILES["fileToUpload"]["name"]);

$conn = mysqli\_connect("localhost","u558220466\_user1","onlookernews123")or die(mysqli\_error());

mysqli\_select\_db($conn,"u558220466\_db1");

$newsid=$NewsId;

$title=$\_POST['title'];

$description=$\_POST['add'];

$time=$\_POST['time'];

$date=$\_POST['date'];

$pic=basename( $\_FILES["fileToUpload"]["name"]);

mysqli\_query($conn,"insert into News values(DEFAULT,'$title','$description','$time','$date','$pic','$status',$rpid)");

echo '<script language="javascript">alert("Saved successfully")</script>';

header("Location:ReporterHome.php");

}

?>

**SplashActivity.java**

package es.esy.onlookernewsdemo;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import es.esy.onlookernewsdemo.onlookernews.R;

import static es.esy.onlookernewsdemo.onlookernews.R.\*;

public class SplashActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(layout.activity\_splash);

Thread timerThread = new Thread() {

public void run() {

try {

sleep(3000);

} catch (InterruptedException e) {

e.printStackTrace();

} finally {

Intent intent = new Intent(SplashActivity.this, TabbedActivity.class);

startActivity(intent);

}

}

};

timerThread.start();

}

@Override

protected void onPause() {

// TODO Auto-generated method stub

super.onPause();

finish();

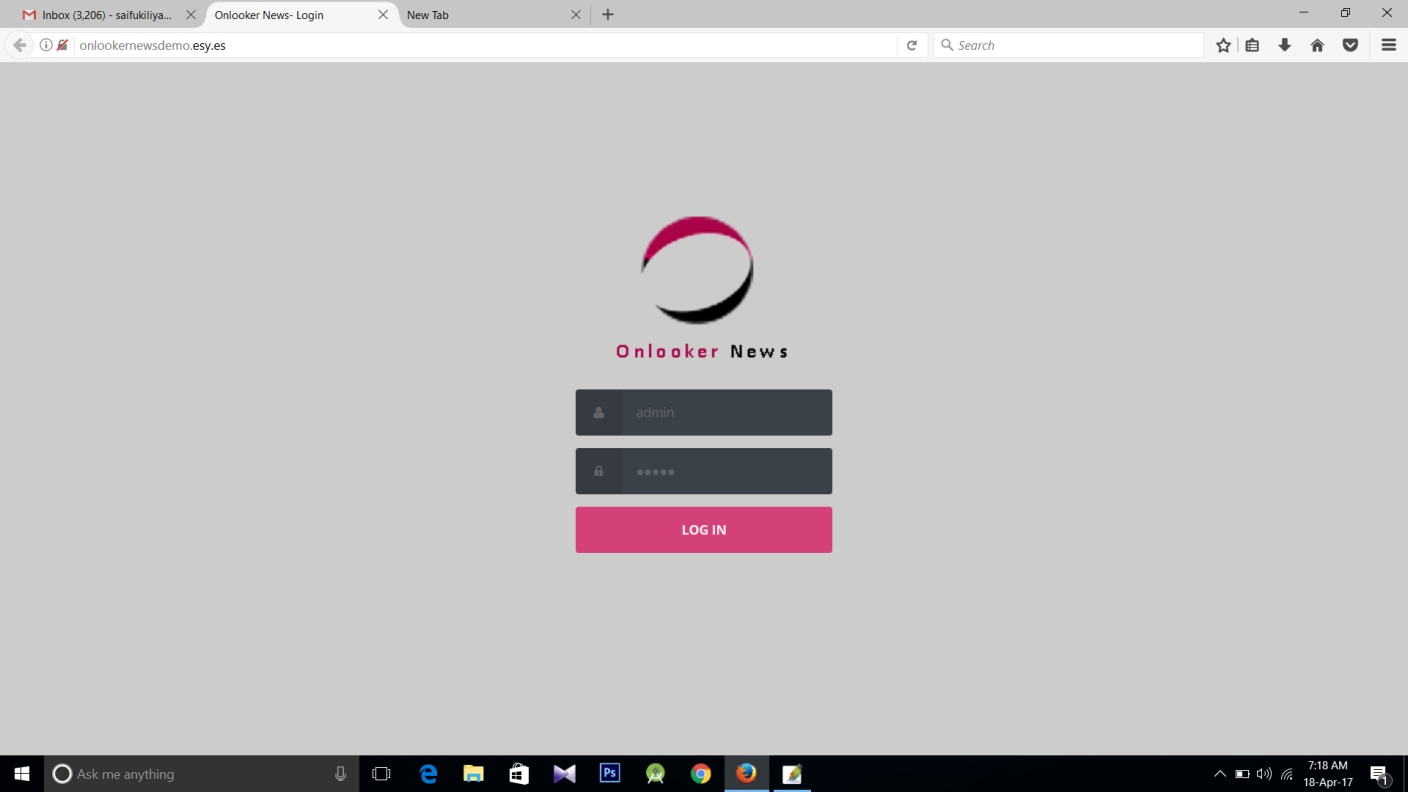
}

}

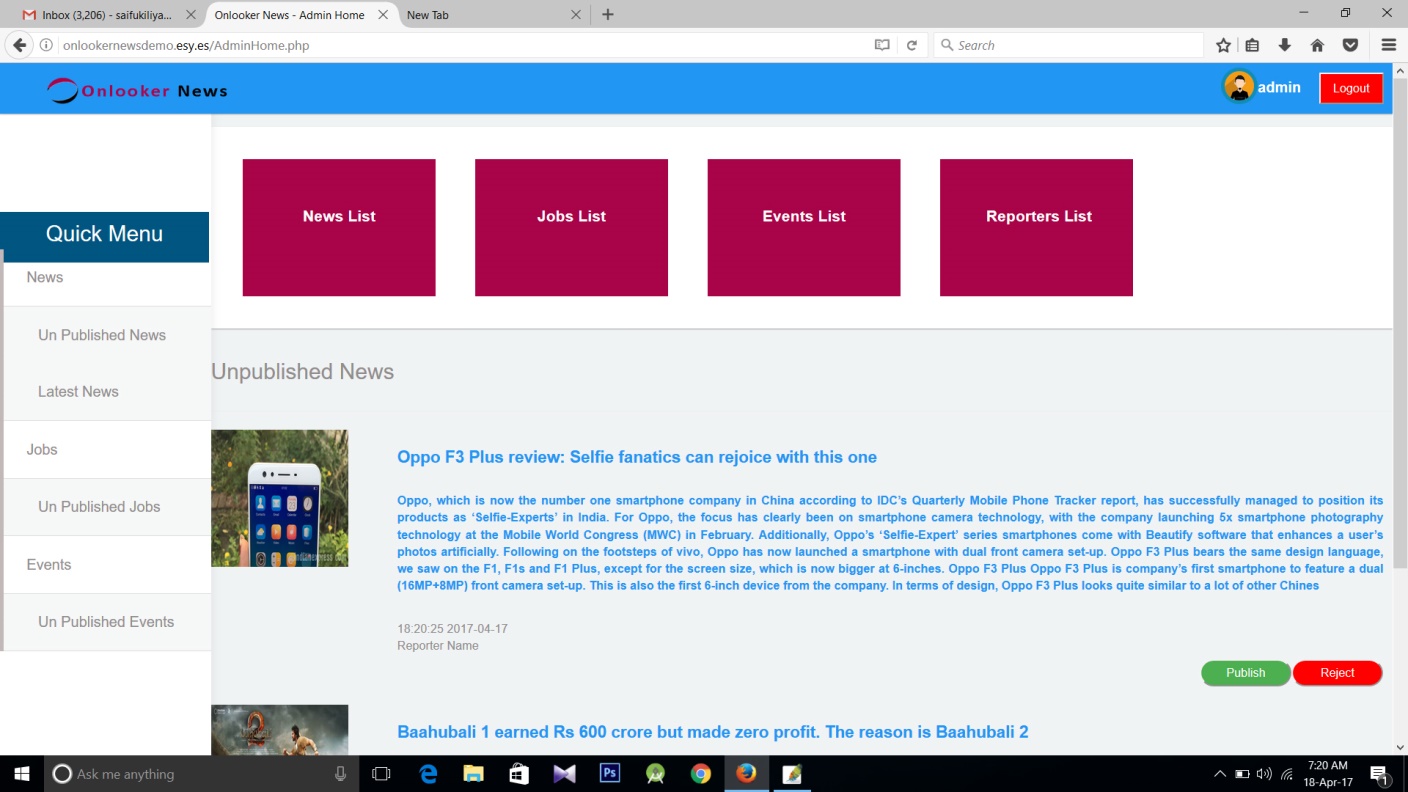
**10.2 APPENDIX B**

**10.2.1 Screen shots**

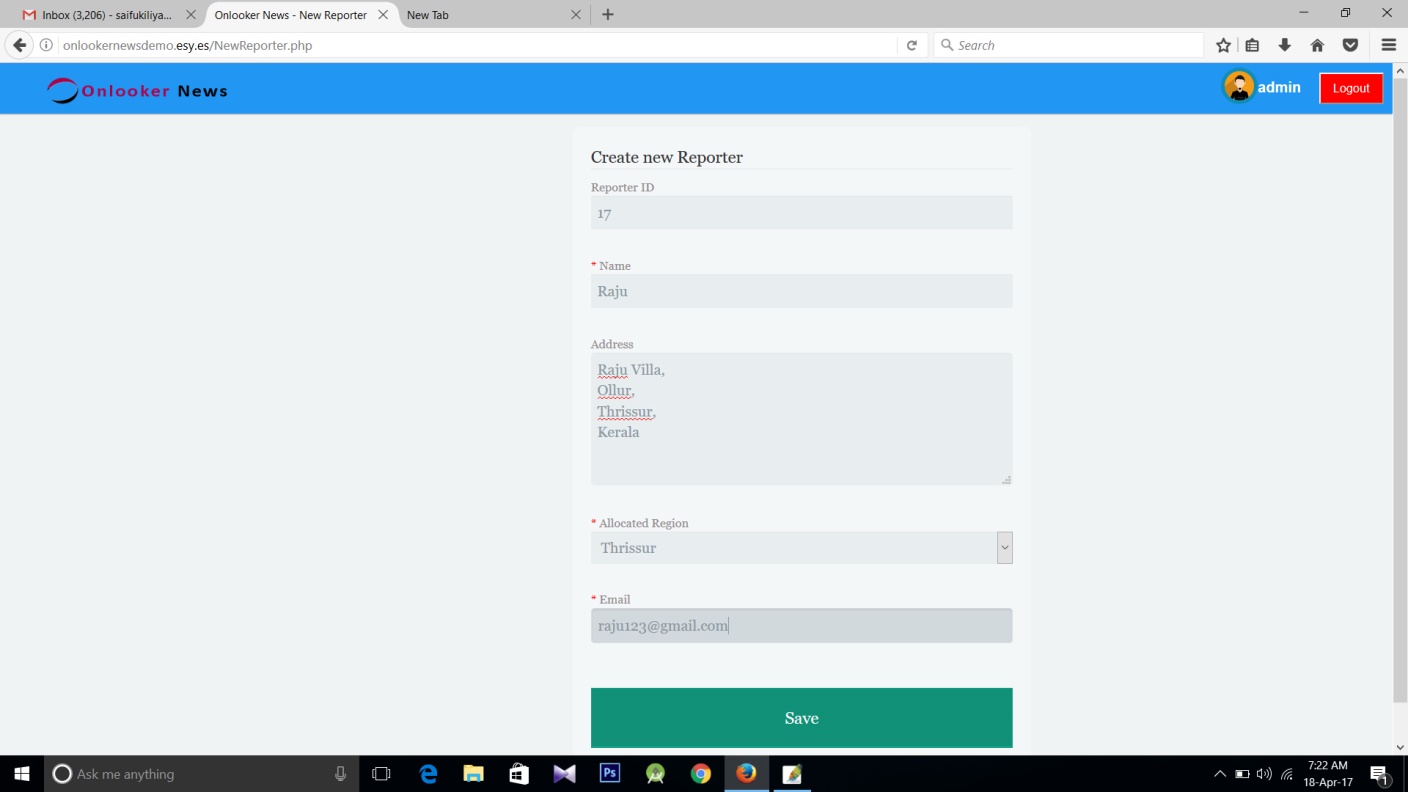
**Admin Login**

****

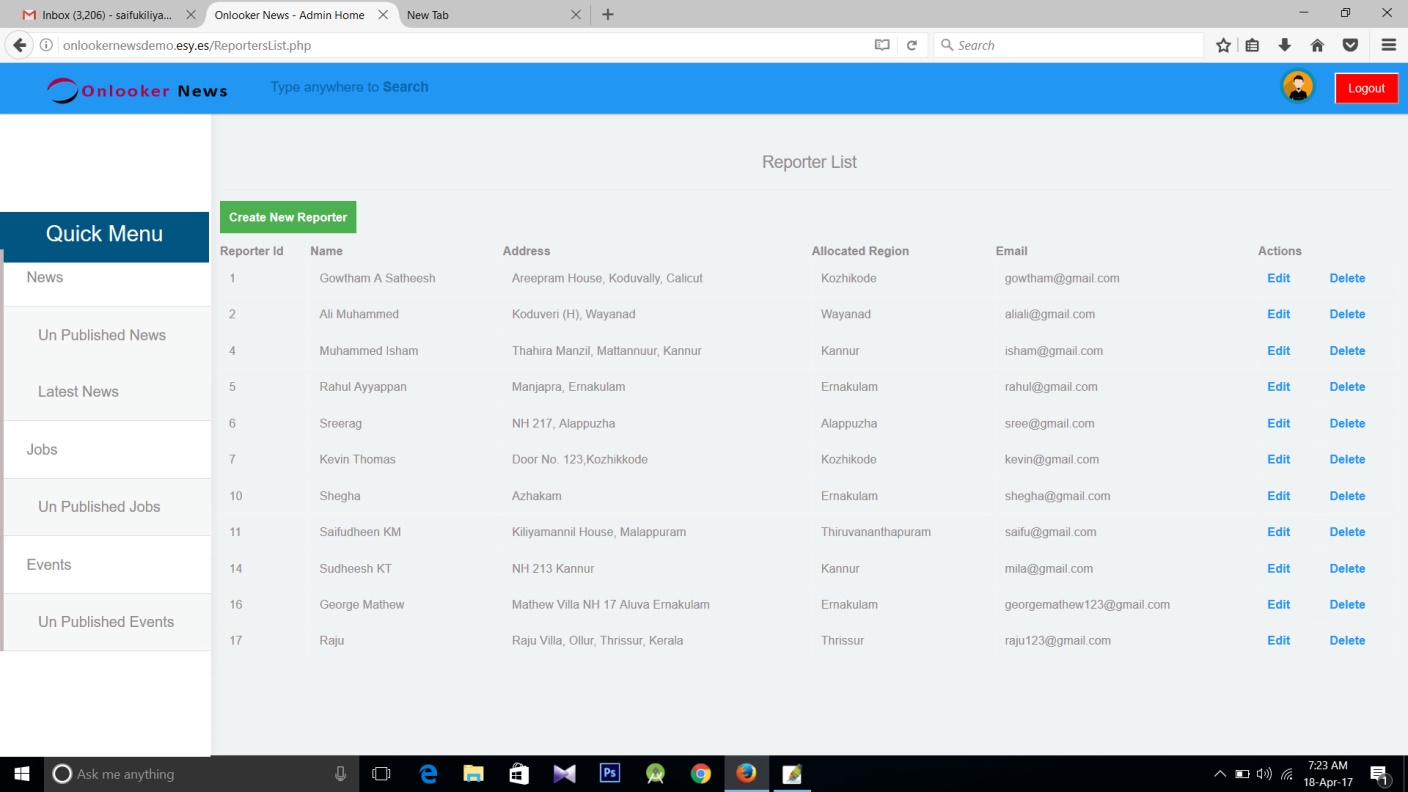
**Admin Home**

****

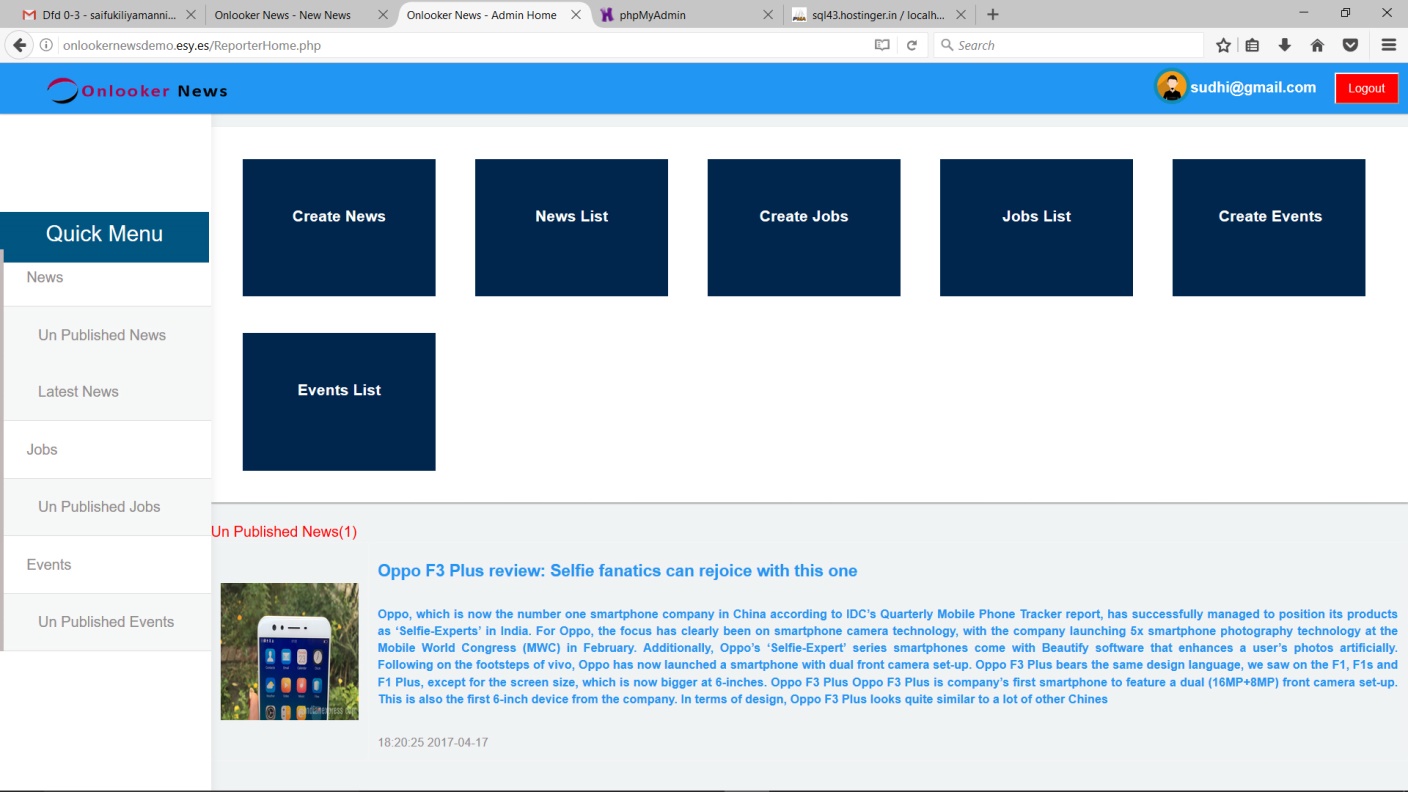
**Reporter Registration**

****

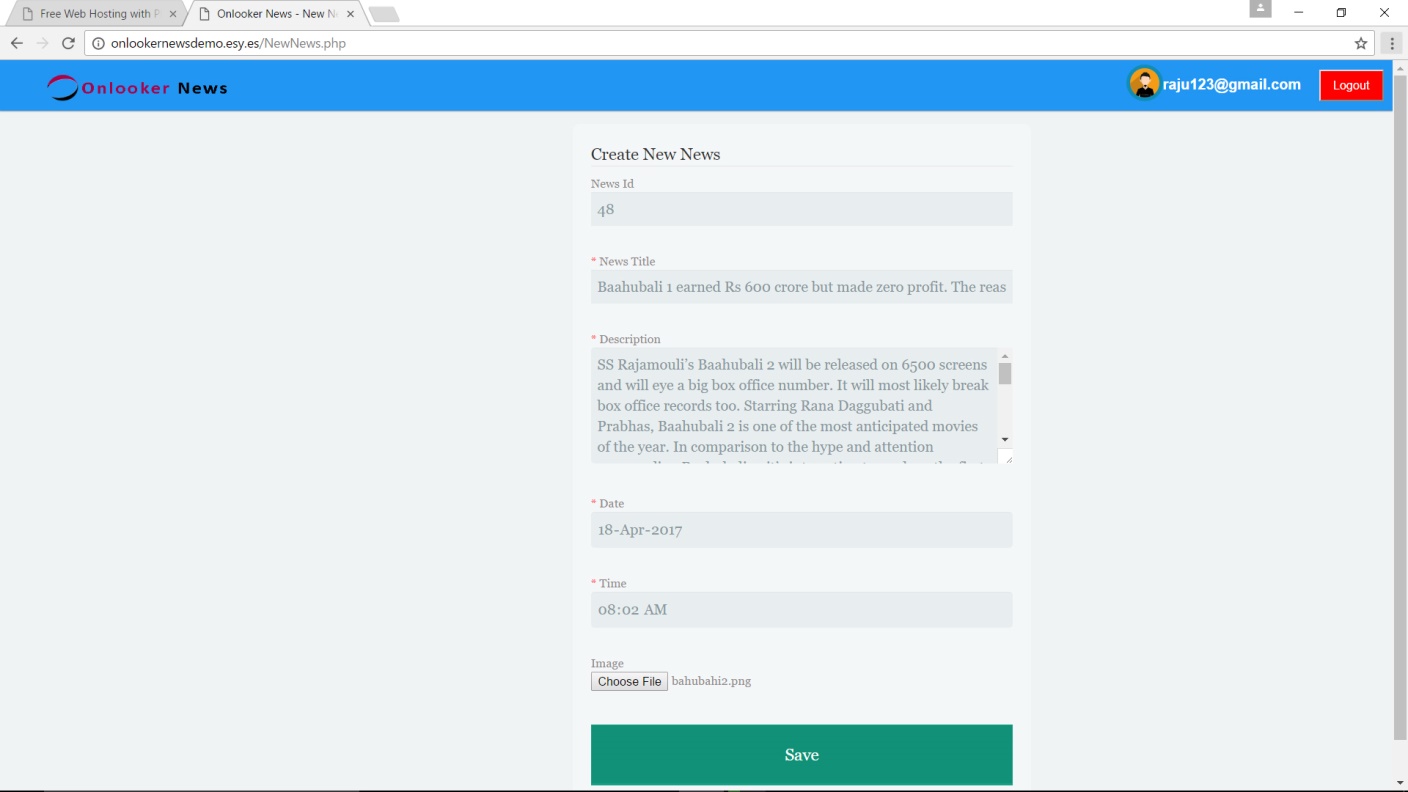
**Reporter List**

****

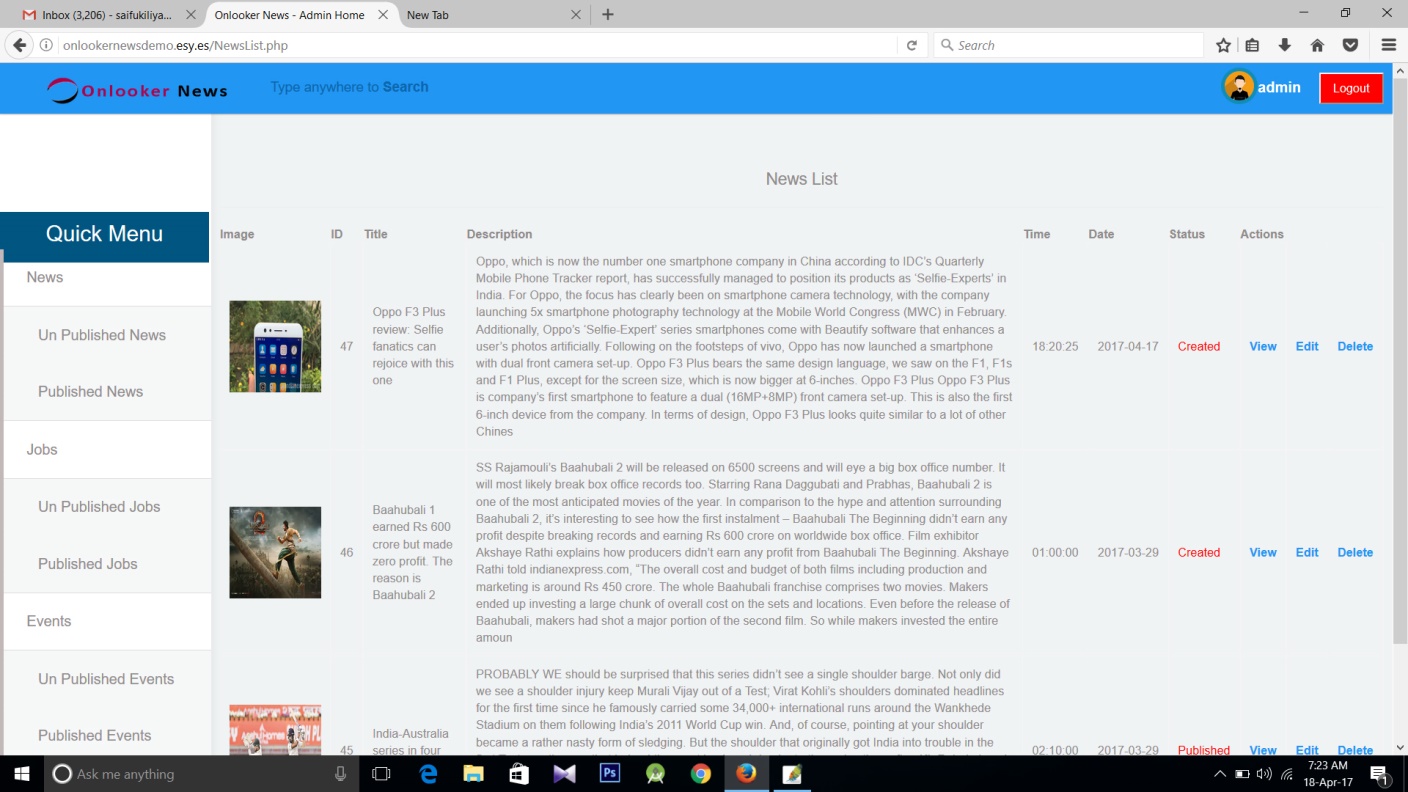
**Reporter Home**

****

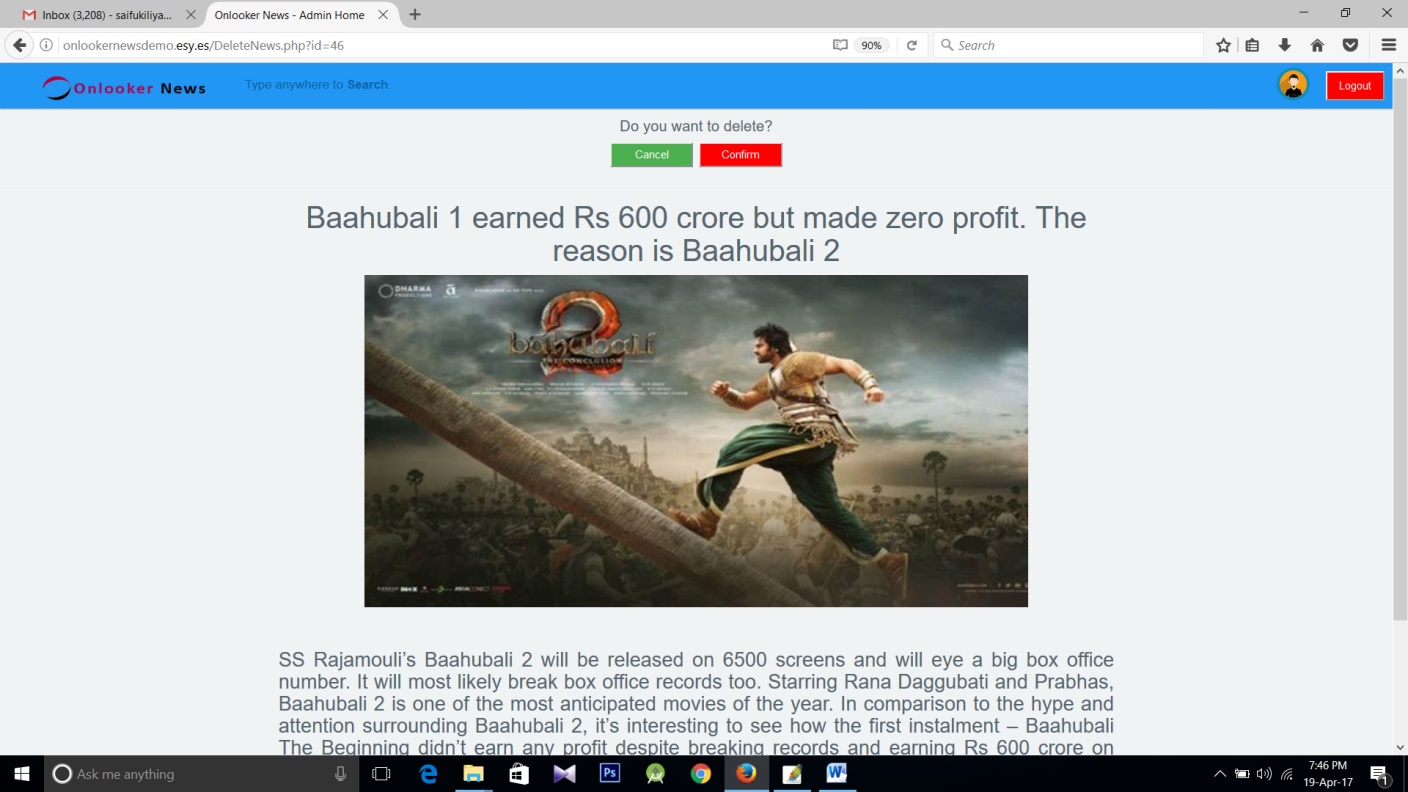
**Create New News**

****

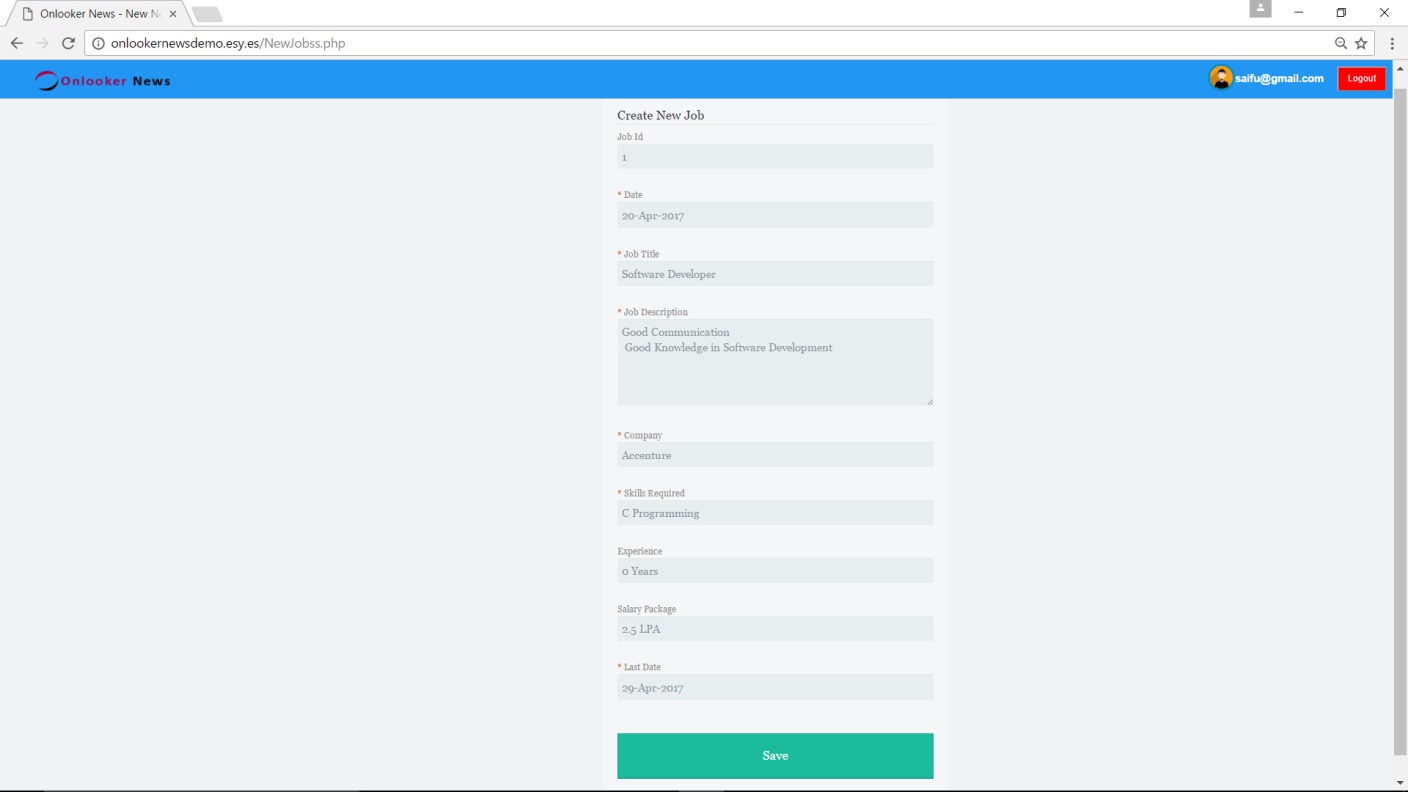
**News List**

****

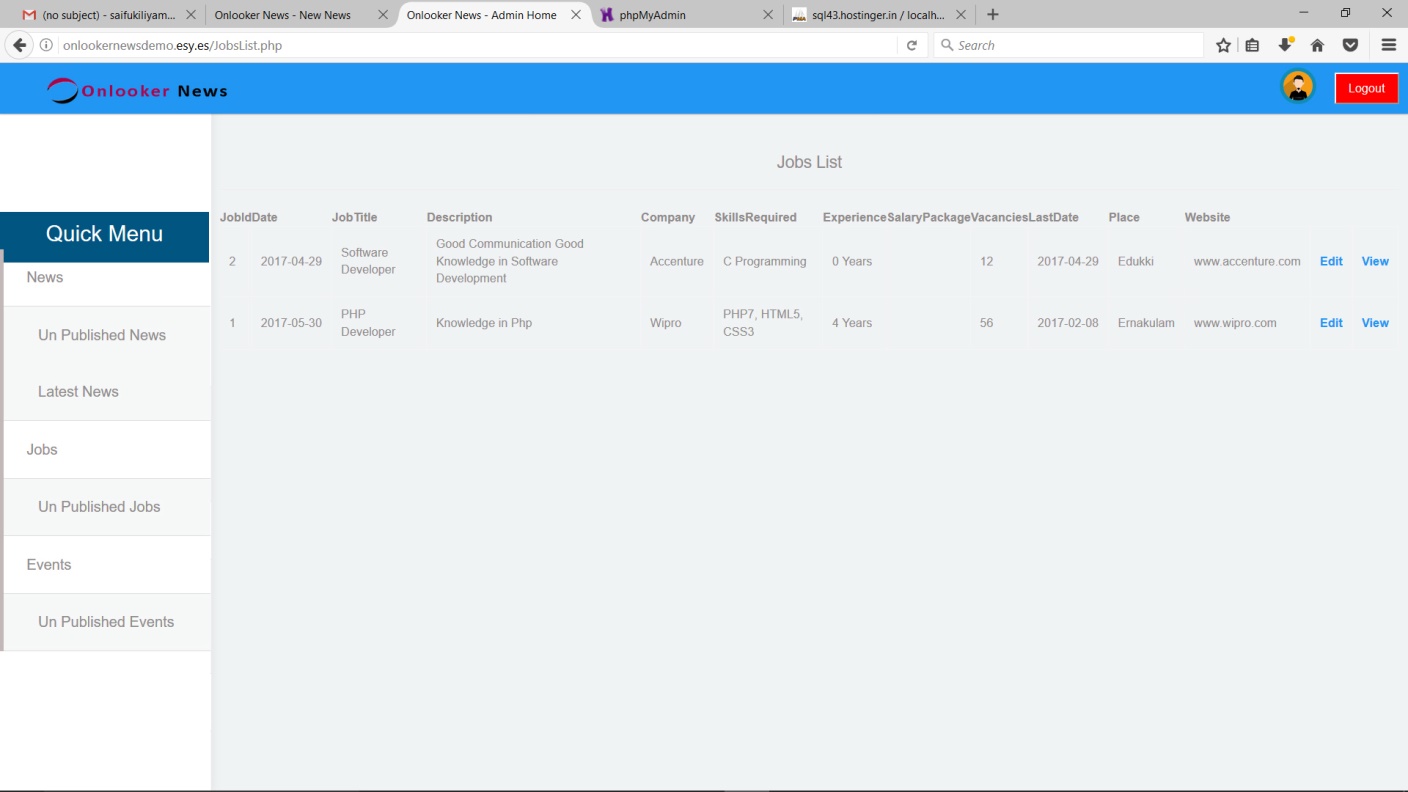
**Delete News**

****

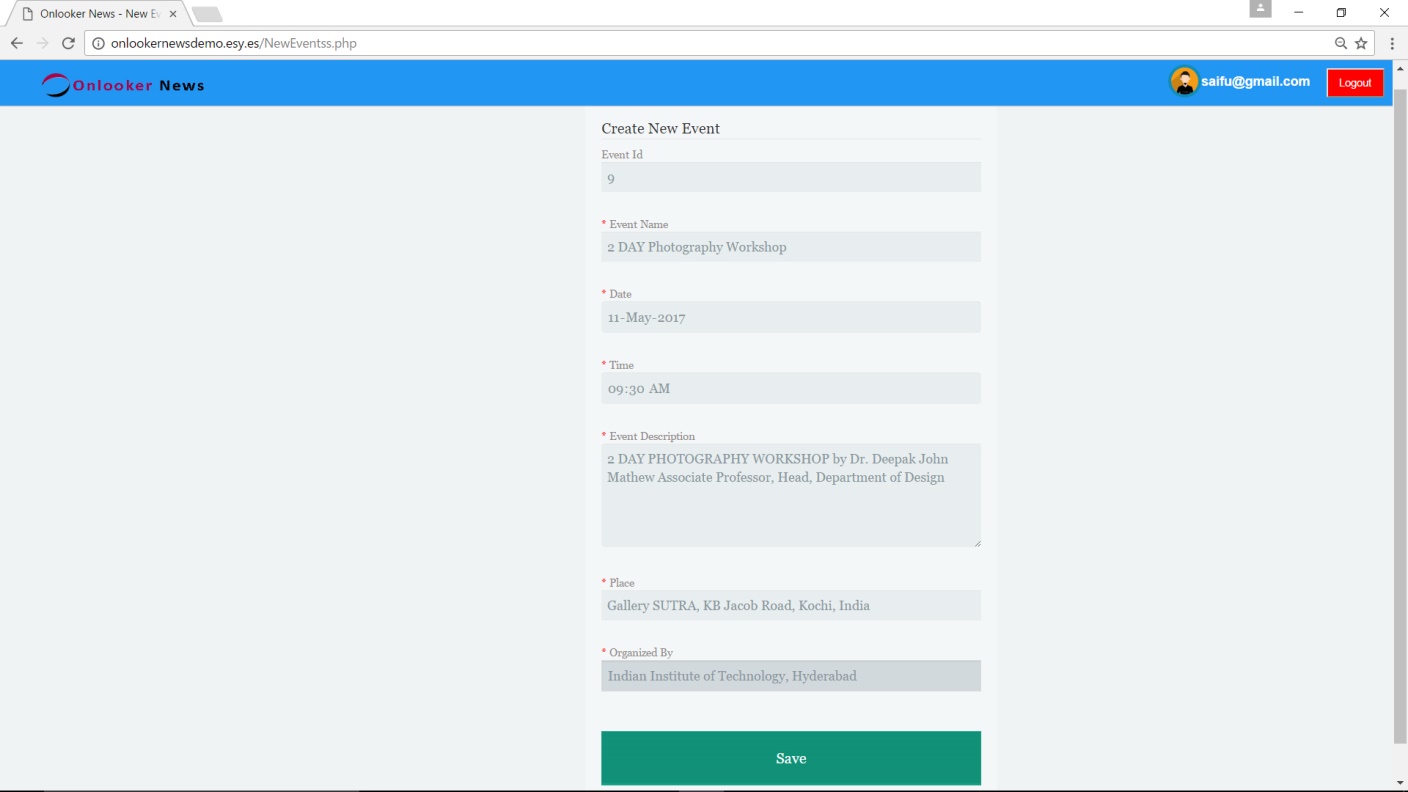
**Create New Jobs**



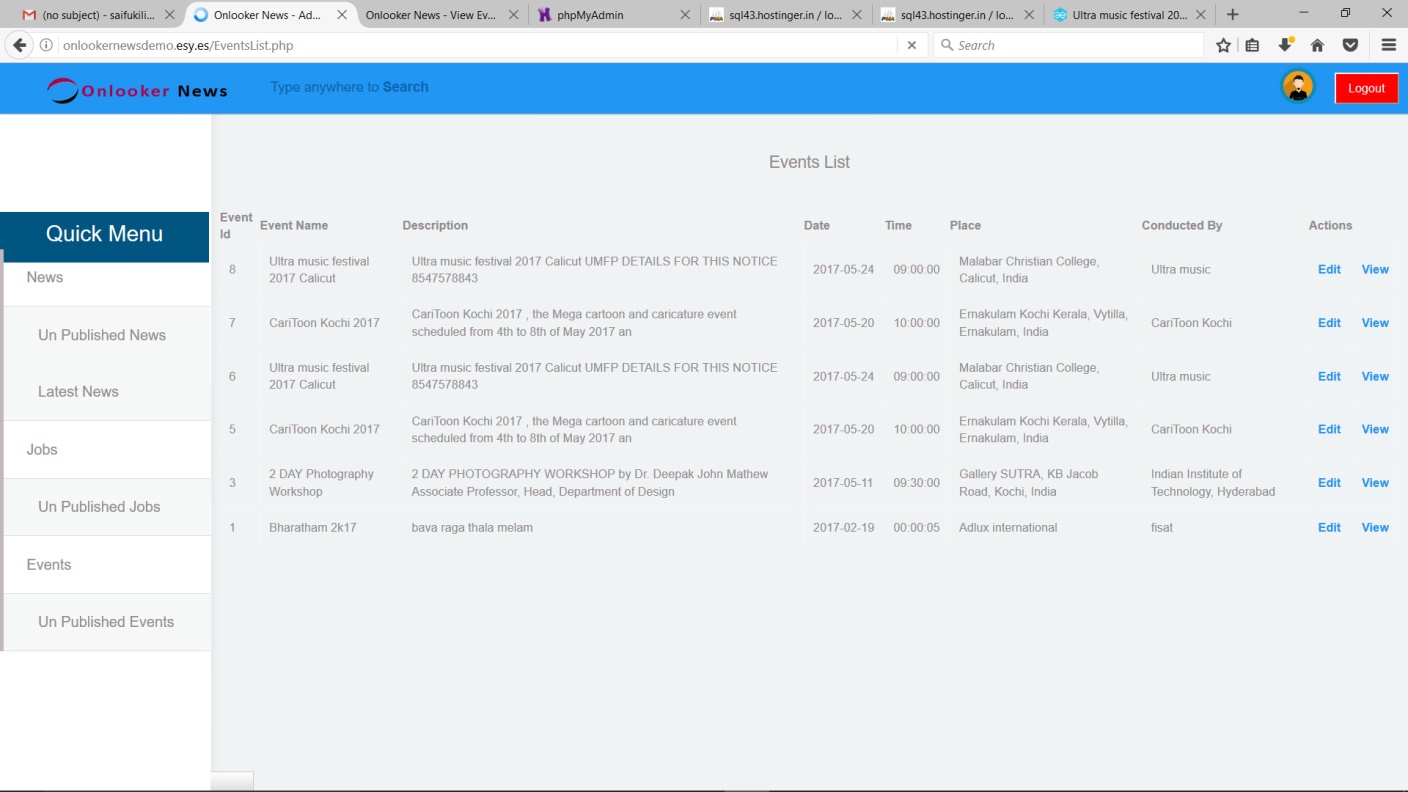
**Jobs List**

****

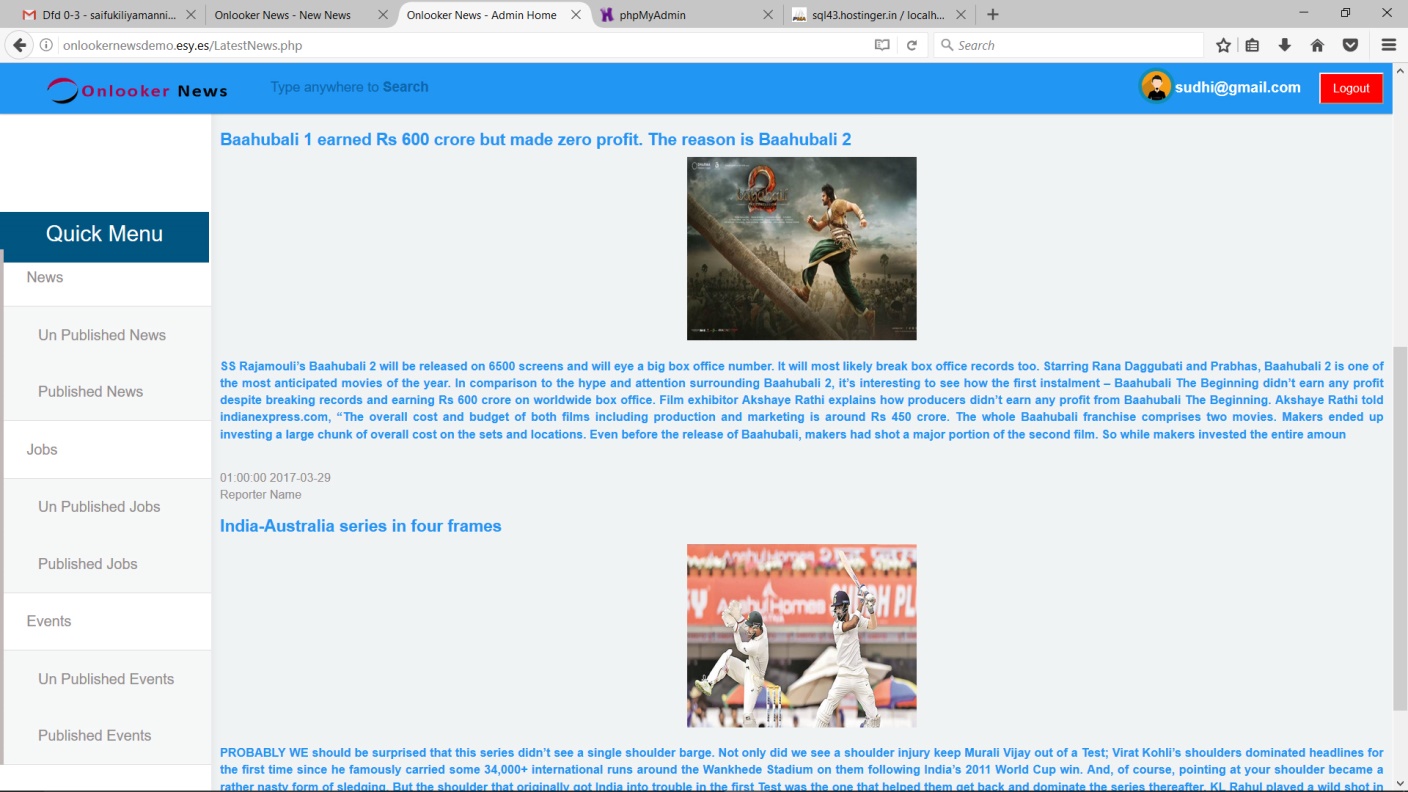
**Create New Events**

****

**Events List**

****

**Latest News**

****

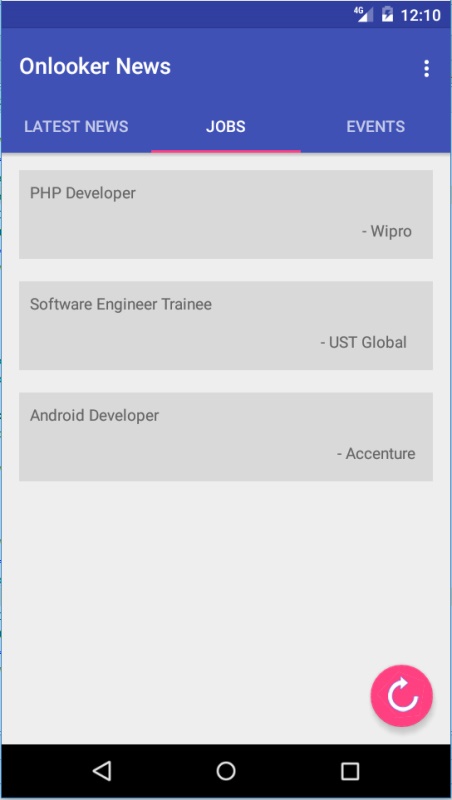
**Splash Screen**

****

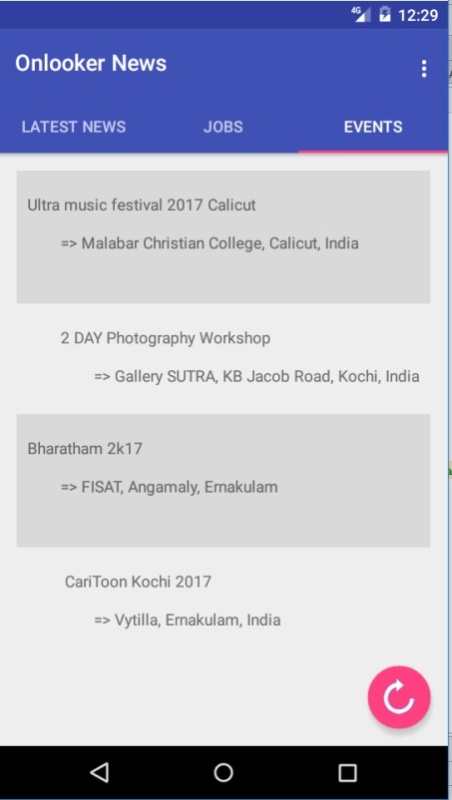
**Latest News**

****

**Job Vacancies**

****

**Upcoming Events**

****

**10.3 APPENDIX C**

**10.3.1 Acronyms**

* DB : Database
* DBMS : Database Management System
* DFD : Data Flow Diagram
* GUI : Graphical User Interface
* SQL : Structured Query Language
* CSS : Cascade Style Sheet

**10.3.2 Bibliography**

**Book referred**

# Android Application Development (With Kitkat Support), Black Book

1. Robin Nixon(2010). Learning PHP,MySQL,JavaScript & CSS
2. Ian Sommerville (2010).Software Engineering

**Web Links Referred**

1. www.hostinger.in
2. www.w3school.com
3. www.developer.android.com