JOB NOTIFIER
MINI PROJECT-2

Mini Project report submitted in partial
fulfillment of requirements
for the award of degree of

Bachelor of Technology
In
Information Technology

By

M.AJAY KARTHIK	**(Reg No:19131A1205)**
A.TARUN TEJA	**(Reg No:19131A1207)**
B.ABHIRAM DORA	**(RegNo:19131A1209)**
G.NAVEEN	**(Reg No:19131A1237)**

Under the esteemed guidance of

Mr.Ch.Rajesh

(Assistant Professor)

Department of Information Technology

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING(AUTONOMOUS)

(Affiliated to JNTU-K, Kakinada)

VISAKHAPATNAM

2022 - 2023

Gayatri Vidya Parishad College of Engineering(Autonomous)

Visakhapatnam

CERTIFICATE

This report on "***Job Notifier"*** is a bonafide record of the mini project - II work submitted

By

M.AJAY KARTHIK	**(Reg No:19131A1205)**
A.TARUN TEJA	**(Reg No:19131A1207)**
B.ABHIRAM DORA	**(RegNo:19131A1209)**
G.NAVEEN	**(Reg No:19131A1237)**

in their VII semester in partial fulfillment of the requirements for the Award of Degree of

Bachelor of Technology
In
Information Technology

During the academic year 2022-2023

Mr.Ch.Rajesh
Assistant Professor,Dept.of IT
Project Guide

Dr. R.V.V Murali Krishna
Head of the Department
Department of Information Technology

DECLARATION

We hereby declare that this industry oriented main project entitled **“JOB NOTIFIER”** is a bonafide work done by us and submitted to **Department of Information Technology G.V.P college of engineering (Autonomous) Visakhapatnam,** in partial fulfillment for the award of the degree of B.Tech is of our own and it is not submitted to any other university or has been published any time before.

PLACE:VISAKHAPATNAM

DATE:

G.NAVEEN (RegNo:19131A1237)
A.TARUN TEJA (RegNo:19131A1207)
B.ABHIRAM DORA(Reg No:19131A1209)
M.AJAY KARTHIK (RegNo:19131A1205)

ACKNOWLEDGEMENT

We would like to express our deep sense of gratitude to our esteemed institute **Gayatri Vidya Parishad College of Engineering (Autonomous),** which has provided us an opportunity to fulfil our cherished desire.

We express our deep sense of Gratitude to **Dr.R.V.V.MURALI KRISHNA, Head of the Department of Information Technology, Gayatri Vidya Parishad** for giving us an opportunity to do the project in college.

We express our sincere thanks to our Principal **Prof. Dr. A. B. KOTESWARARAO** for his encouragement to us during the course of this project, giving us a chance to explore and learn new technologies in the form of mini project.

We express our profound gratitude and our deep indebtedness to our guide **Mr.CH.RAJESH** whose valuable suggestions, guidance and comprehensive assistance helped us a lot in realizing our present project.

We also thank our coordinator, **Mr.D.NAGA TEJ** for the kind suggestions and guidance for the successful completion of our project work.

Finally we would also like to thank all the members of the teaching and non-teaching staff of the Information Technology Department for all their support in completion of our project.

M.AJAY KARTHIK	**19131A1205**
A.TARUN TEJA	**19131A1207**
B.ABHIRAM DORA	**19131A1209**
G.NAVEEN	**19131A1237**

ABSTRACT

Internet is a source of live data that is constantly updating with data of almost any field .Having tools that can automatically detect these updates and select that information that we are interested in,will have utmost importance now-a-days.Web scraping will helps to do it.Web scraping is a method of extracting and restricting the data from web pages.

This application can be used on different dynamic websites i.e.., job portals and automate the repeatitive tasks.It will reduce the manual work of going to a website and searching for new job multiple times.As the application collects the data from job portal and send it to the user.

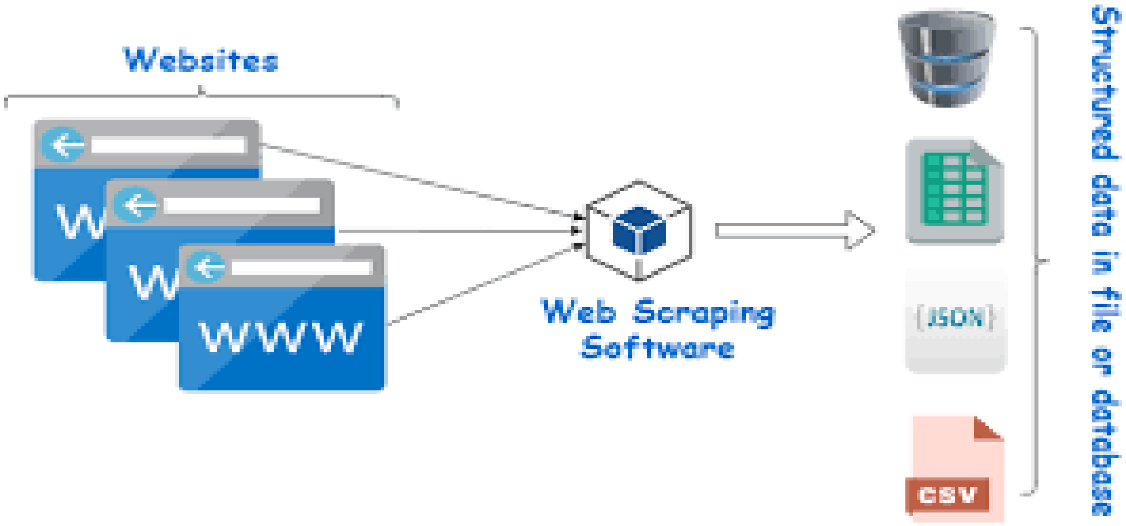
This application will find new openings on job portals and alert the user by providing required information.

CONTENTS

INTRODUCTION

WEB SCRAPPING:

Web Scraping is a web technique for extracting data from the web and turning unstructured data on the web (including HTML formats) into structured data that you can store to your local computer or database.



1.1 OBJECTIVE:

The main objective of the application is to find new openings on job portals and alert the user by providing required information.

1.2 SCOPE:

This application can be used on different dynamic websites i.e.., job portals and automate the repeatitive tasks.It will reduce the manual work of going to a website and searching for new job multiple times.As the application collects the data from job portal and send it to the user.

2.SRS DOCUMENT

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfill all stakeholders (business, users) needs.

2.1 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a system or its component, where a function of a system or its component is described as a specification of behavior between inputs and outputs.

The functional requirements in our project are:

INPUT:

User should provide:

Job Type-Job title,keywords, or company

Location-City,state,or pin code

PROCESS:

- Scraping the data from job portal and transforming the data.
- Alerting the user by sending the required information.

OUTPUT:

New Job Openings are sent to user.

2.2 NON FUNCTIONAL REQUIREMENTS

A Non-Functional Requirement (NFR) defines the quality attribute of a software system.They judge the software system based on Security, Performance,Maintainability,Reliabiltyand other non-functional standards that are critical to success of the software system.

Non-Functional Requirements are:

Performance:

Response Time : The system provides acknowledgment in just a few seconds.

Capacity : The system needs to support at least 100 people at once.

User-Interface : The user interface acknowledges within five seconds.

Maintainability:

Back-Up : The system offers the efficiency for data back up.

Reliability:

Availability : The system is available all the time.

Cache:

We can cache URLs that are frequently accessed by the users. The UGS servers, before making a query to the database, may check if the cache has the desired URL. Then it does not need to make the query again.

2.3 SOFTWARE REQUIREMENTS

Software requirements deal with defining software resource requirements that need to be installed on a computer to provide optimal functioning of an application.

Operating System: Windows7/XP/8/10

TECHNOLOGIES USED: JAVASCRIPT , NODE JS

IDE: VS CODE

2.4 HARDWARE REQUIREMENTS

Processor: Intel dual core i3/i5/i7

RAM: 4 GB

3.ALGORITHM ANALYSIS

3.1 EXISTING SYSTEM

- Web scraping technique is used that helps to collect the data (job openings data) anf transfer in either ,CSV or json file to help you better understand the information you have gathered.
- It is done manually which is long and tedious process.

3.2 PROPOSED SYSTEM

- This application will automate the repetitive tasks.
- It will reduce the manual work of going to a website and searching for a new job openings multiple times.
- Adding notifier and periodic scheduling that will alert the user when the job applications are open after specified interval of time

3.3 FEASIBILITY STUDY

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

3.3.1 Economical Feasibility

This is a very important aspect to be considered while developing a project. We decided the technology based on the minimum possible cost factor.All hardware and software cost has to be borne by the organization.Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for the system.

3.3.2 TECHNICAL FEASIBILITY

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of frontend and backend platform.

3.3.3 OPERATIONAL FEASIBILITY

No doubt the proposed system is fully GUI based and is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, proper training has been conducted to let know the essence of the system so that they feel comfortable with the new system.

3.3 COST BENEFIT ANALYSIS

Software cost comprises a small percentage of overall computer-based system cost. There are a number of factors, which are considered, that can affect the ultimate cost of the software such as human, technical, Hardware and Software availability etc. The main point that was considered during the cost estimation of project was its sizing. In spite of complete software sizing, function point and approximate lines of code were also used to "size" each element of the Software and their costing.The cost estimation done for Project also depend upon the baseline metrics collected from existing system and these were used in conjunction with estimation variables to develop cost and effort projections.

We have basically estimated this project mainly on two bases -

1) **Effort Estimation** - This refers to the total man-hours required for the development of the project. It even includes the time required for doing documentation and user manual.

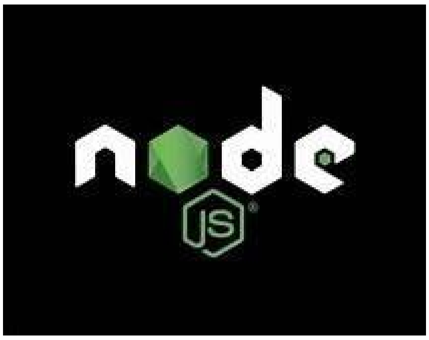
2) **Hardware Required Estimation**- This includes the cost of the PCs and the hardware cost required for development of this project.

SOFTWARE DESCRIPTION

4.1 JAVASCRIPT

JavaScript often abbreviated as **JS**, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). JavaScript engines were originally used only in web browsers, but they are now core components of other software systems, most notably servers and a variety of applications.

4.2Node.js



As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications.**Node.js** is an open-source, cross-platform, back-end

JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

5.PROJECT DESCRIPTION

5.1 PROBLEM DEFINITION

- To find new openings on job portals and alert the user by providing required information.

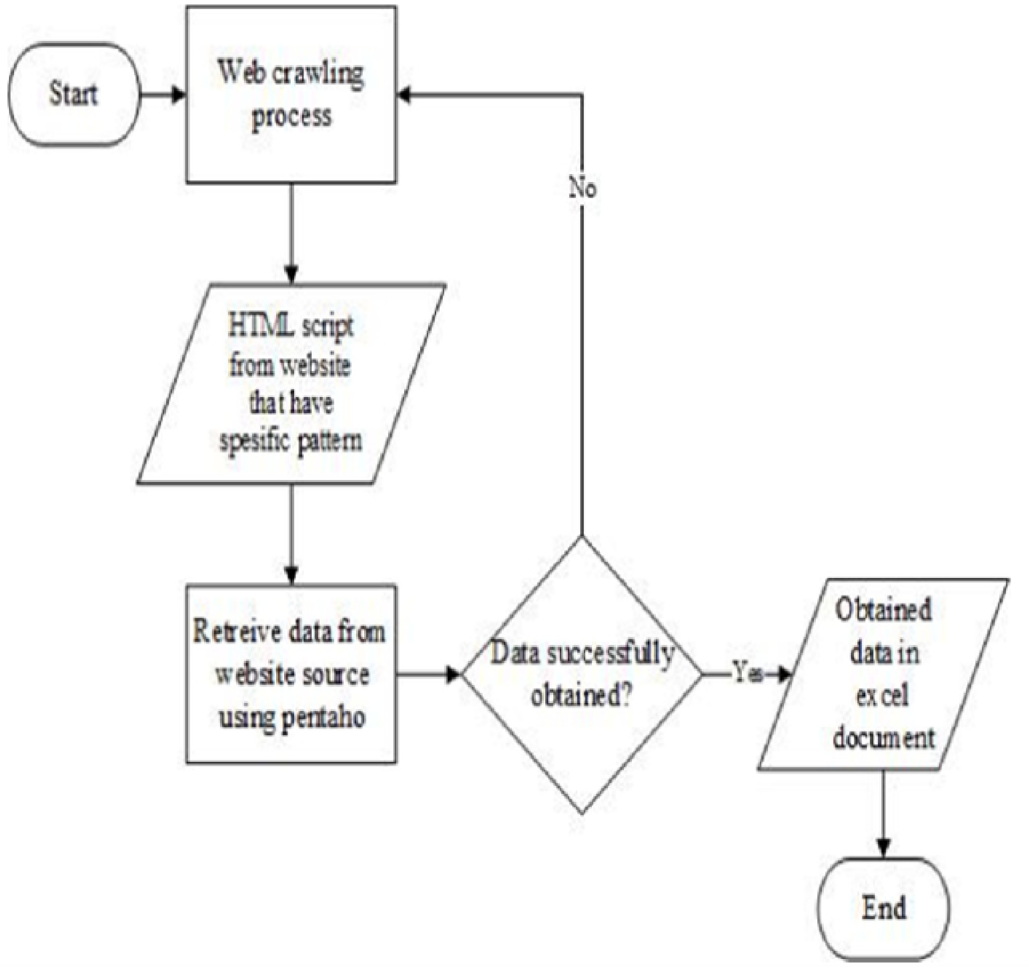
5.2 PROJECT OVERVIEW

- User specifies the job type and location
- Scrap the data from the job portal and transform it structured data.
- This data is sent to the user using Gmail and this process is repeated continously after a

Specified amount of time.

6.SYSTEM DESIGN

6.1 SYSTEM ARCHITECTURE


Start
Web crawling process
No
HTML script from website that have spesific pattern
Retreive data from website source using pentaho
Data successfully obtained?
Yes
Obtained data in excel document
End

6.2 INTRODUCTION TO UML

The system to be developed is best designed using UML i.e. Unified Modeling Language. The Unified Modeling Language includes a set of graphics notation techniques to create visual models of object-oriented intensive systems. UML is a visual language for specifying, constructing, and documenting the artifacts of software- intensive systems.

Complex software designs difficult for you to describe with text alone can readily be conveyed through diagrams using UML. You can use UML with all processes throughout the development life cycle and across different implementations.

A model is a simplification of reality. We build models so that we can better understand the system we are developing. Through modeling, we achieve four aims. They are:

1. Models help us to visualize a system as it is or as we want it to be.
2. Models permit us to specify the structure or behavior of a system.

6.3 BUILDING BLOCKS OF UML:

The vocabulary of the UML encompasses three kinds of building blocks:

1. Things
2. Relationships
3. Diagrams
   - Things are the abstractions that are first-class citizens in a model;
   - Relationships tie these things together;
   - Diagrams group interesting collections of things;

- Things in the UML:

There are four kinds of things in the UML:

1. Structural things
2. Behavioral things
3. Grouping things
4. Annotational things

6.3.1 Structural Things:

Structural things are the nouns of UML models. These are the mostly static parts of a model, representing elements that are either conceptual or physical. In all, there are seven kinds of structural things.

- A class is a description of a set of objects that share the same attributes, operations, relationships, and semantics. A class implements one or more interfaces. Graphically, a class is rendered as a rectangle, usually including its name, attributes, and operations.

- An interface is a collection of operations that specify a service of a class or component. An interface therefore describes the externally visible behavior of that element. An interface might represent the complete behavior of a class or component or only a part of that behavior.

- Collaboration defines an interaction and is a society of roles and other elements that work together to provide some cooperative behavior that's bigger than the sum of all the elements. Therefore, collaborations have structural, as well as behavioral,dimensions. Graphically, collaboration is rendered as an ellipse with dashed lines, usually including only its name.

- Use case is a description of set of sequence of actions that a system performs that yields an observable result of value to a particular actor. A use case is used to structure the behavioral things in a model. Graphically, a use case is rendered as an ellipse with solid lines, usually including only its name.

- An active class is a class whose objects own one or more processes or threads and therefore can initiate control activity. An active class is just like a class except that its objects represent elements whose behavior is concurrent with other elements. Graphically, an active class is rendered just like a class, but with heavy lines, usually including its name, attributes, and operations.

- A component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces. Graphically, a component is rendered as a rectangle with tabs, usually including only its name.

- A node is a physical element that exists at run time and represents a computational resource, generally having at least some memory and, often, processing capability. A set of components may reside on a node and may also migrate from node to node. Graphically, a node is rendered as a cube, usually including only its name.

6.3.2 Behavioral Things:

- Behavioral things are the dynamic parts of UML models. These are the verbs of a model, representing behavior over time and space. In all, there are two primary kinds of behavioral things.
- An interaction is a behavior that comprises a set of messages exchanged among a set of objects within a particular context to accomplish a specific purpose.
- A state machine is a behavior that specifies the sequences of states an object or an interaction goes through during its lifetime in response to events, together with its responses to those events. A state machine involves a number of other elements, including states, transitions (the flow from state to state), events (things that trigger a transition), and activities (the response to a transition). Graphically, a state is rendered as a rounded rectangle, usually including its name and its substrates, if any.

6.3.3 Grouping Things:

- Grouping things are the organizational parts of UML models. These are the boxes into which a model can be decomposed. In all, there is one primary kind of grouping thing, namely, packages.
- A package is a general-purpose mechanism for organizing elements into groups.
- Structural things, behavioral things, and even other grouping things may be placed in a package. Unlike components, a package is purely conceptual. Graphically, a package is rendered as a tabbed folder, usually including only its name & its contents.

6.3.4 Annotational Things:

- Annotational things are the explanatory parts of UML models. These are the comments you may apply to describe, illuminate, and remark about any element in a model. There is one primary kind of Annotational thing, called a note.

- A note is simply a symbol for rendering constraints and comments attached to an element or a collection of elements. Graphically, a note is rendered as a rectangle with a dog-eared corner, together with a textual or graphical comment.

Relationships in the UML:

There are four kinds of relationships in the UML:

1. Dependency
2. Association
3. Generalization
4. Realization

These relationships are the basic relational building blocks of the UML. We use them to write well-formed models.

- A dependency is a semantic relationship between two things in which a change to one thing may affect the semantics of the other thing (the dependent thing).

- An association is a structural relationship that describes a set of links, a link being a connection among objects.

- A generalization is a specialization/generalization relationship in which objects of the specialized element (the child) are substitutable for objects of the generalized element (the parent).

- A realization is a semantic relationship between classifiers, wherein one classifier specifies a contract that another classifier guarantees to carry out.

Diagrams in the UML:

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships). You draw diagrams tovisualize a system from different perspectives, so a diagram is a projection into a system. UML includes nine diagrams.

- A **class diagram** shows a set of classes, interfaces, and collaborations and their relationships. These diagrams are the most common diagram found in modeling object-oriented systems. Class diagrams address the static design view of a system. Class diagrams that include active classes address the static process view of a system

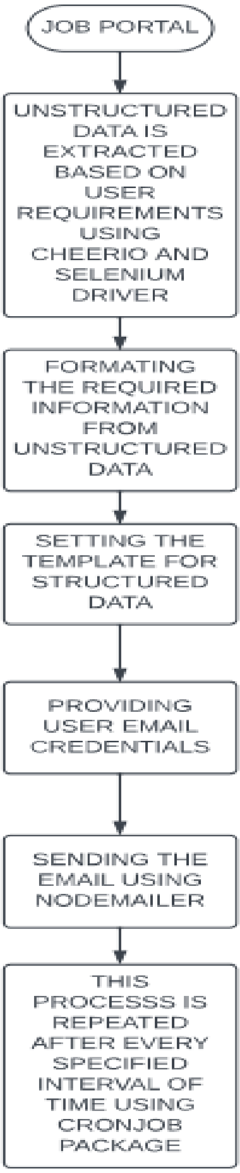
- An **object diagram** shows a set of objects and their relationships. Object diagrams represent static snapshots of instances of the things found in class diagrams. These diagrams address the static design view or static process view of a system as do class diagrams, but from the perspective of real or prototypical cases.

- A **use case diagram** shows a set of use cases and actors (a special kind of class) and their relationships. Use case diagrams address the static use case view of a system. These diagrams are especially important in organizing and modeling the behavior of a system.

- Both **sequence diagrams** and **collaboration diagrams** are kinds of interaction diagrams. An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them.

Interaction diagrams address the dynamic view of a system. A sequence diagram is an interaction diagram that emphasizes the time-ordering of messages; a collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Sequence diagrams and collaboration diagrams are isomorphic, meaning that we can take one and transform it into the other.

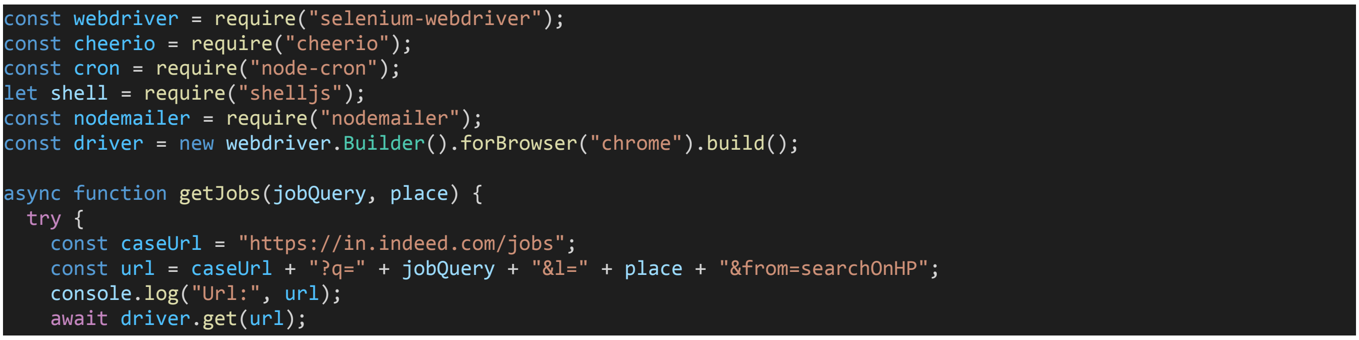
- A **state chart diagram** shows a state machine, consisting of states, transitions, events, and activities. State chart diagrams address the dynamic view of a system


JOB PORTAL
UNSTRUCTURED DATA IS EXTRACTED BASED ON USER REQUIREMENTS USING CHEERIO AND SELENIUM DRIVER
FORMATING THE REQUIRED INFORMATION FROM UNSTRUCTURED DATA
SETTING THE TEMPLATE FOR STRUCTURED DATA
PROVIDING USER EMAIL CREDENTIALS
SENDING THE EMAIL USING NODEMAILER
THIS PROCESSS IS REPEATED AFTER EVERY SPECIFIED INTERVAL OF TIME USING CRONJOB PACKAGE

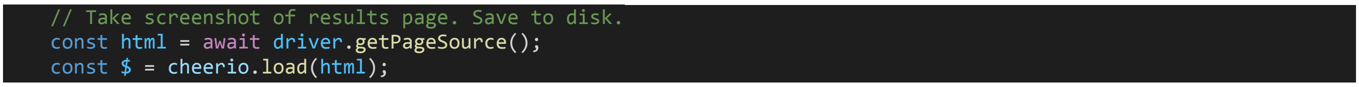
7.DEVELOPMENT

7.1 SAMPLE CODE

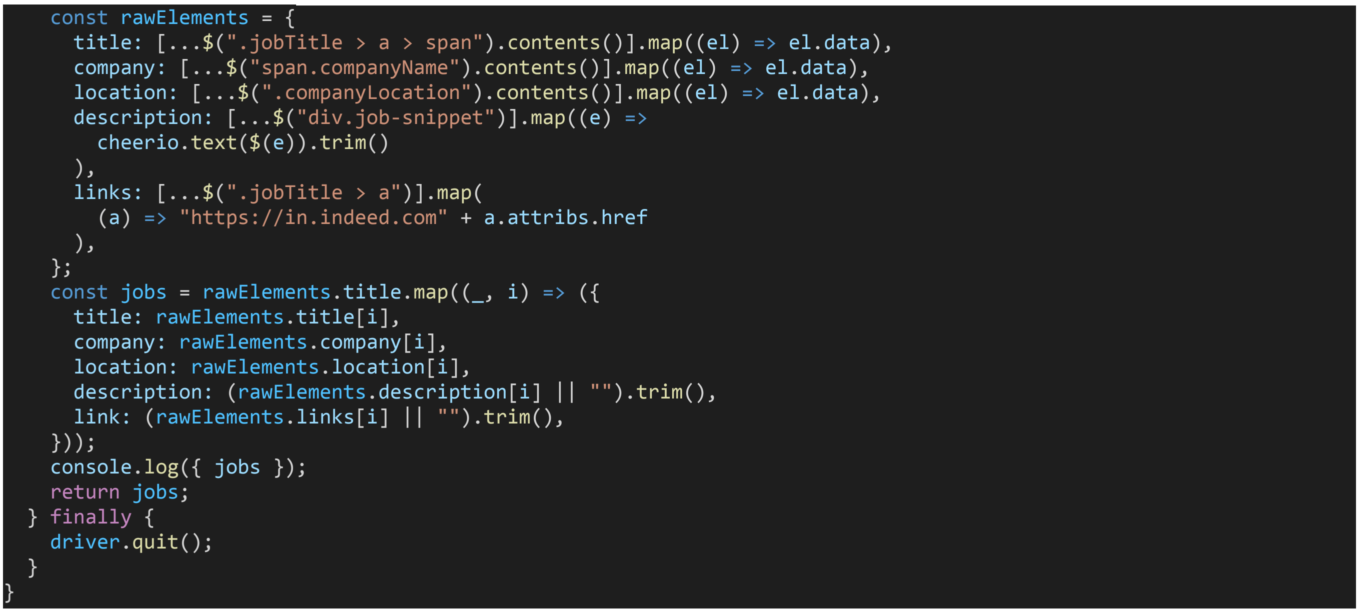
App.js:



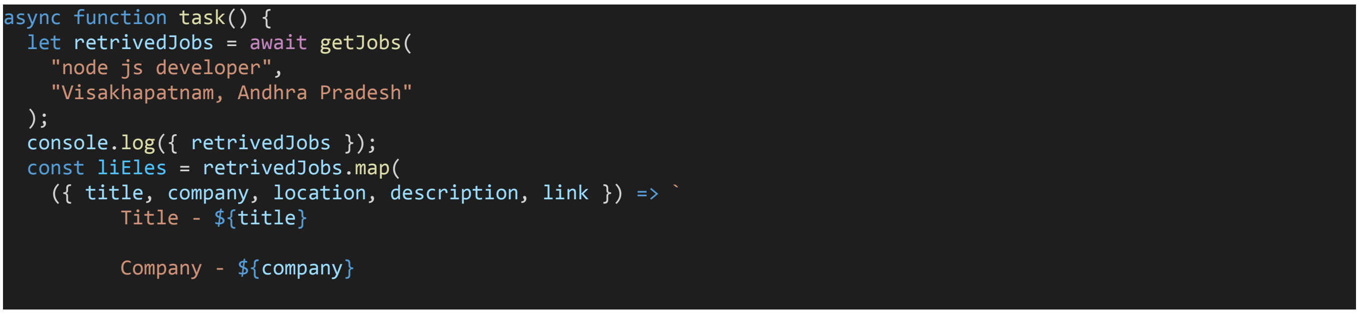
```
const webdriver = require("selenium-webdriver");
const cheerio = require("cheerio");
const cron = require("node-cron");
let shell = require("shelljs");
const nodemailer = require("nodemailer");
const driver = new webdriver.Builder().forBrowser("chrome").build();

async function getJobs(jobQuery, place) {
  try {
    const caseUrl = "https://in.indeed.com/jobs";
    const url = caseUrl + "?q=" + jobQuery + "&l=" + place + "&from=searchOnHP";
    console.log("Url:", url);
    await driver.get(url);
```



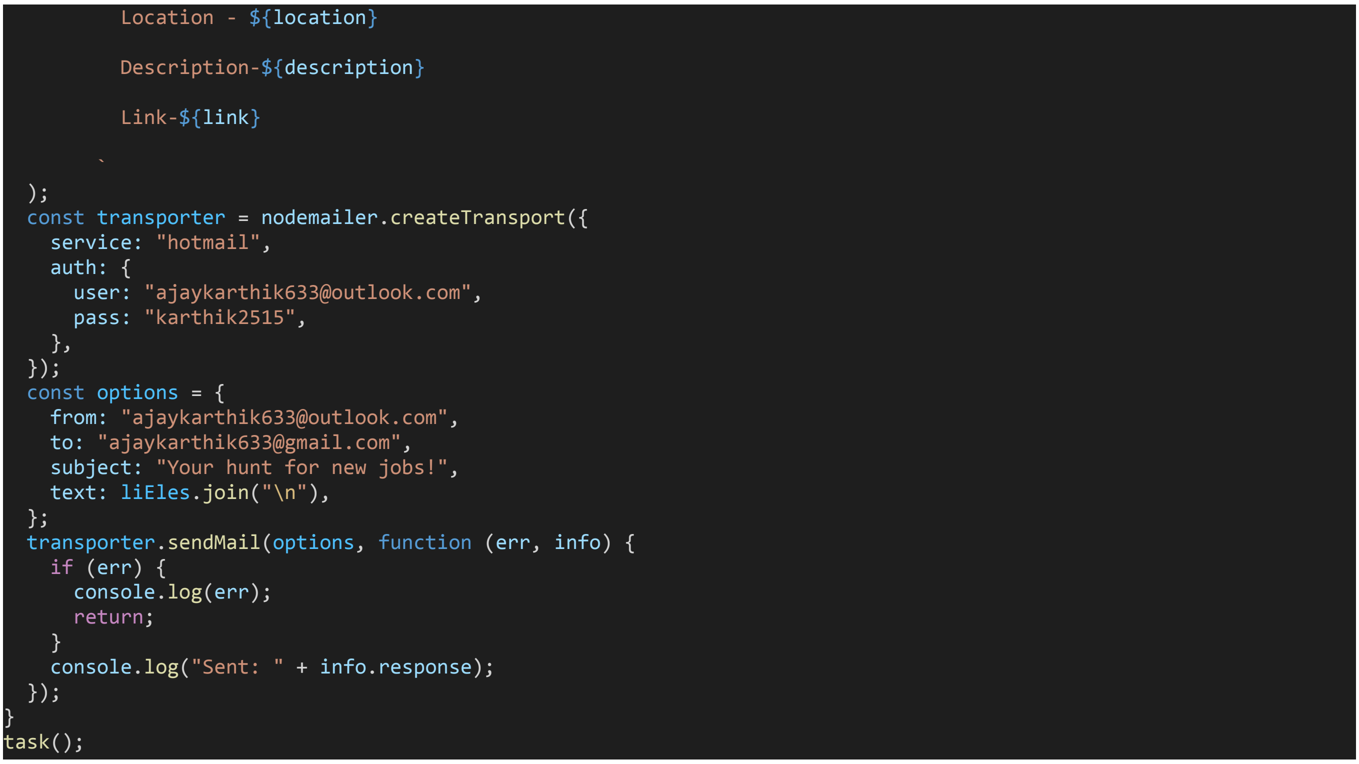
```
    // Take screenshot of results page. Save to disk.
    const html = await driver.getPageSource();
    const $ = cheerio.load(html);
```



```
    const rawElements = {
      title: [...$(".jobTitle > a > span").contents()].map((el) => el.data),
      company: [...$("span.companyName").contents()].map((el) => el.data),
      location: [...$(".companyLocation").contents()].map((el) => el.data),
      description: [...$("div.job-snippet")].map((e) =>
        cheerio.text($(e)).trim()
      ),
      links: [...$(".jobTitle > a")].map(
        (a) => "https://in.indeed.com" + a.attribs.href
      ),
    };
    const jobs = rawElements.title.map((_, i) => ({
      title: rawElements.title[i],
      company: rawElements.company[i],
      location: rawElements.location[i],
      description: (rawElements.description[i] || "").trim(),
      link: (rawElements.links[i] || "").trim(),
    }));
    console.log({ jobs });
    return jobs;
  } finally {
    driver.quit();
  }
}
```



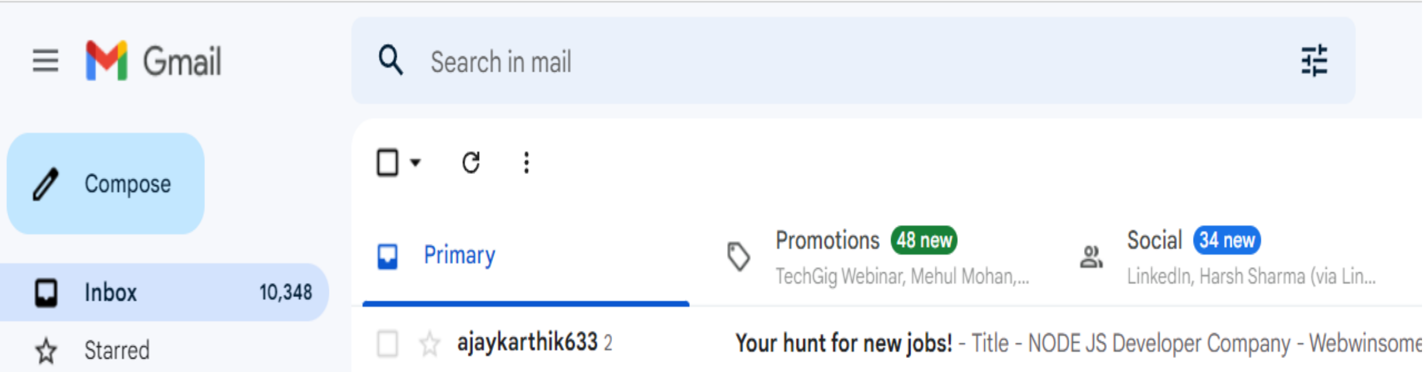
```
async function task() {
  let retrievedJobs = await getJobs(
    "node js developer",
    "Visakhapatnam, Andhra Pradesh"
  );
  console.log({ retrievedJobs });
  const liEles = retrievedJobs.map(
    ({ title, company, location, description, link }) => `
        Title - ${title}

        Company - ${company}
```



```
          Location - ${location}

          Description-${description}

          Link-${link}

        `
  );
  const transporter = nodemailer.createTransport({
    service: "hotmail",
    auth: {
      user: "ajaykarthik633@outlook.com",
      pass: "karthik2515",
    },
  });
  const options = {
    from: "ajaykarthik633@outlook.com",
    to: "ajaykarthik633@gmail.com",
    subject: "Your hunt for new jobs!",
    text: liEles.join("\n"),
  };
  transporter.sendMail(options, function (err, info) {
    if (err) {
      console.log(err);
      return;
    }
    console.log("Sent: " + info.response);
  });
}
task();
```

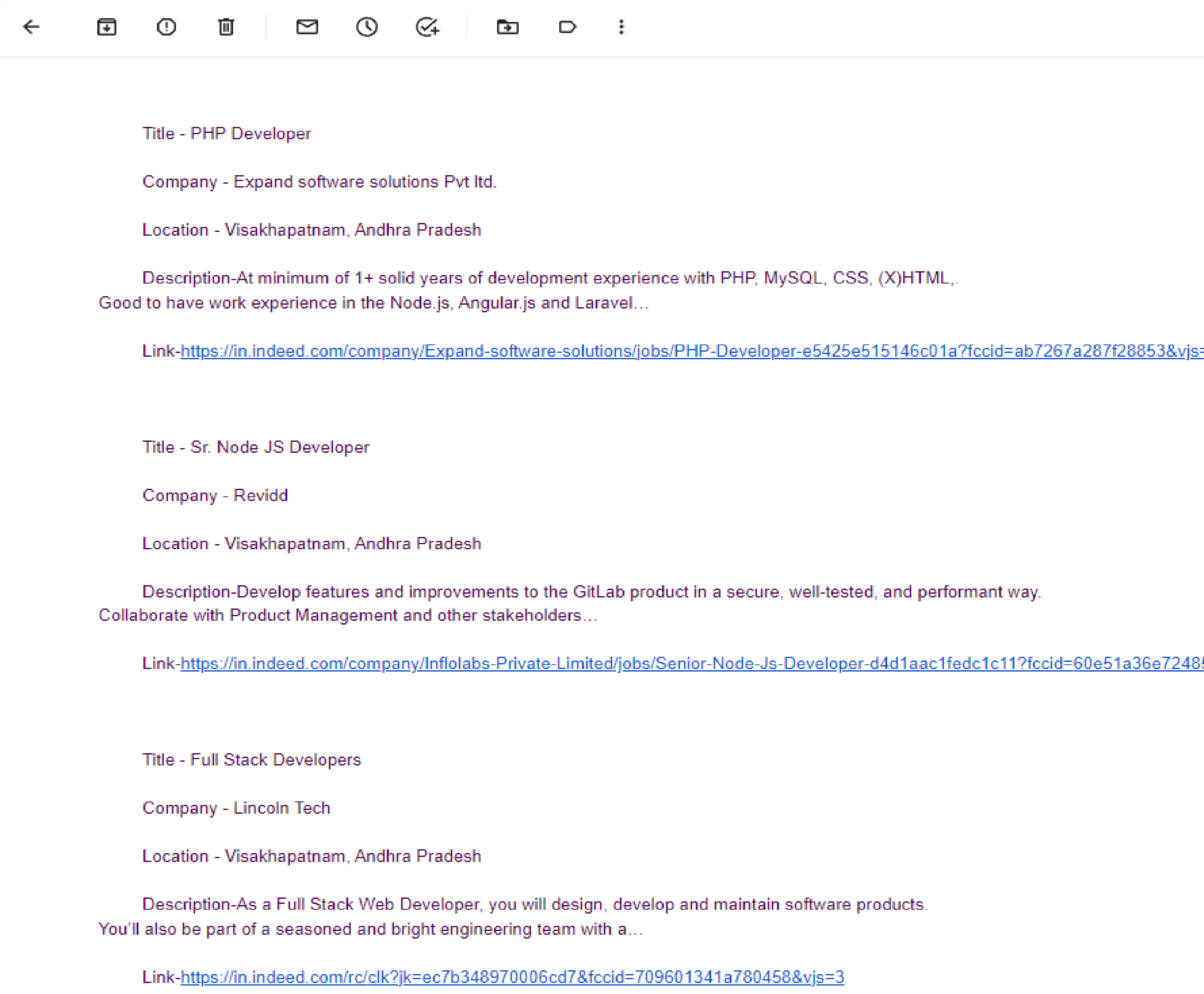
7.2 INPUT OUTPUT SCREENS :

EMAIL NOTIFICATION:



ajaykarthik633 2 **Your hunt for new jobs!** - Title - NODE JS Developer Company - Webwinsome

INFORMATION REGRADING NEW JOB OPENINGS:



Title - PHP Developer

Company - Expand software solutions Pvt ltd.

Location - Visakhapatnam, Andhra Pradesh

Description-At minimum of 1+ solid years of development experience with PHP, MySQL, CSS, (X)HTML,.
Good to have work experience in the Node.js, Angular.js and Laravel…

Link-https://in.indeed.com/company/Expand-software-solutions/jobs/PHP-Developer-e5425e515146c01a?fccid=ab7267a287f28853&vjs=

Title - Sr. Node JS Developer

Company - Revidd

Location - Visakhapatnam, Andhra Pradesh

Description-Develop features and improvements to the GitLab product in a secure, well-tested, and performant way.
Collaborate with Product Management and other stakeholders…

Link-https://in.indeed.com/company/Inflolabs-Private-Limited/jobs/Senior-Node-Js-Developer-d4d1aac1fedc1c11?fccid=60e51a36e7248

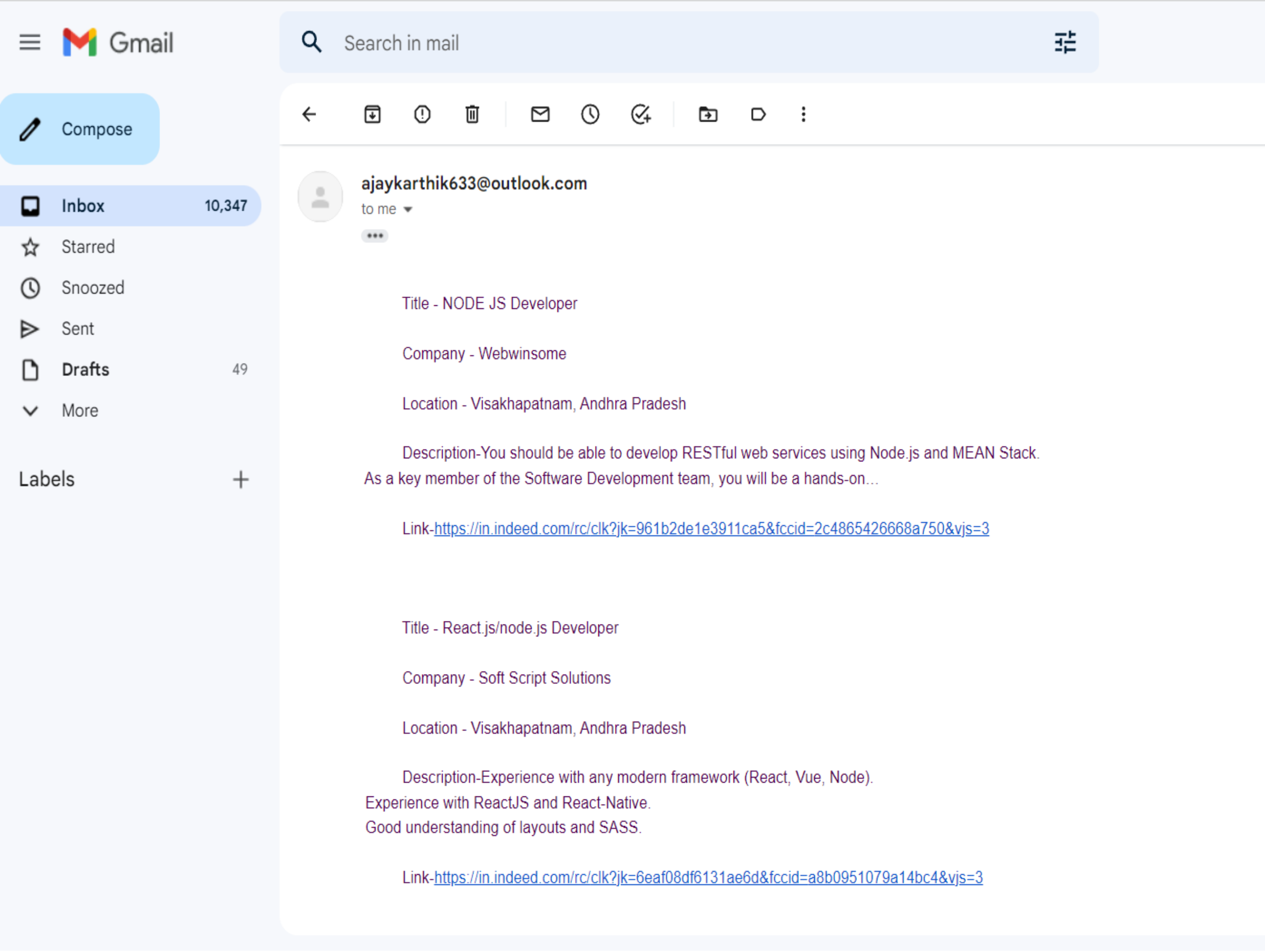
Title - Full Stack Developers

Company - Lincoln Tech

Location - Visakhapatnam, Andhra Pradesh

Description-As a Full Stack Web Developer, you will design, develop and maintain software products.
You'll also be part of a seasoned and bright engineering team with a…

Link-https://in.indeed.com/rc/clk?jk=ec7b348970006cd7&fccid=709601341a780458&vjs=3



ajaykarthik633@outlook.com
to me

Title - NODE JS Developer

Company - Webwinsome

Location - Visakhapatnam, Andhra Pradesh

Description-You should be able to develop RESTful web services using Node.js and MEAN Stack.
As a key member of the Software Development team, you will be a hands-on...

Link-https://in.indeed.com/rc/clk?jk=961b2de1e3911ca5&fccid=2c4865426668a750&vjs=3

Title - React.js/node.js Developer

Company - Soft Script Solutions

Location - Visakhapatnam, Andhra Pradesh

Description-Experience with any modern framework (React, Vue, Node).
Experience with ReactJS and React-Native.
Good understanding of layouts and SASS.

Link-https://in.indeed.com/rc/clk?jk=6eaf08df6131ae6d&fccid=a8b0951079a14bc4&vjs=3

8.TESTING

8.1 INTRODUCTION

The development of software includes a series of productive activities and testing is an important activity of them. This phase is a critical element of software quality assurance and represents the ultimate review of specification, coding and testing.

The main objectives of testing are as follows:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

Testing can be done in different ways. Some of the types of testing are mentioned below. The main purpose of any type of test is to systematically uncover different classes of errors and do so with a minimum amount of time and effort.

Types of testing:

- Unit testing
- Integration testing
- Regression testing
- System testing
- Performance Testing etc..

Testing can be done manually or by testing tools. There are several testing tools for different software.

Unit Testing: It is a method by which individual units of source code, sets of one or more computer modules together with associated control data, usage procedures and operating procedures, are tested to determine if they are fit for use. Unit testing focuses verification effort on the smallest unit of software design-the software component or module. The unit test is white-box oriented. The unit testing is implemented in every module of student attendance management System. By giving correct manual input to the system, the data's are stored in database and retrieved. If we want required module to access input or get the output, it should be checked for the accessibility and result is to be displayed else error message should be displayed.

Integration Testing: It is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

Regression Testing: Regression testing is any type of software testing that seeks to uncover new software bugs, or regressions, in existing functional and non-functional areas of a system after changes such as enhancements, patches or configuration changes, have been made to them.

System Testing: System testing of software or hardware is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements. System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system.

Performance Testing: Performance testing is designed to test the run-time performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as white-box tests are conducted. This project reduces manual work and helps in getting students attendance status much faster. No extra time is required waiting for the results, as soon as the correct data is entered the results will be displayed in a few milliseconds.

Alpha Testing: Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developer's site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

Beta Testing: Beta testing comes after alpha testing and can be considered as a form of external user acceptance testing. Versions of the software, known as beta versions, are released to a limited outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes beta versions are made available to the open public to increase the feedback field to a maximal number of future users.

Each module can be tested using the following two strategies:

1. Internal program logic is exercised using ―White box ‖ test case design techniques.
2. Software requirements are exercised using ―Black box ‖ test case design techniques.

In both cases, the intent is to find the maximum number of errors with the Minimum amount of effort and time.

Black box Testing:

In this strategy some test cases are generated as input conditions that execute all functional requirements for the program. This testing is used to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structure or external database access

- Performance errors
- Initialization and termination errors

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

White box Testing:

In this testing, test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It is a method of testing software tests internal structures or workings of an application as opposed to its functionality (i.e. Black box testing). In white box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths the code and determine appropriate outputs. This is analogous to testing nodes in a circuit. White box Testing can be applied at the unit, integration, system levels of the software testing process. Although traditional testers tended to think of White box testing as being one at the unit level.

If the username or password is incorrect.

9.CONCLUSION

We have developed this scraping model which will rescue at that _point where a lot meaningful time can be saved.The entire process of job _scraping as well as job searching Manual becomes faster. processes get replaced by automated processes.Candidates are just a few clicks away to get connected.This application will be helpful in finding new openings on _job portals and alert the user by providing required information.

10. FUTURE SCOPE

-There is ample scope of enhancement and adding functionalities to this application.

-This application can be extended to send automated interview scheduling acceptance/rejection of resume.

-The application can have a job recommendation system based on frequent search result of different users.

-There can be a feedback or review section for the application.

-The application can be more scalable by extending the search functionality based on country , city or area.

-The User Interface can be made more attractive and user friendly.

11.BIBLIOGRAPHY

1.Vivek Kumar Sehgal1, “Job Portal-A Web Application for Geographically Distributed Multiple Clients”, 1 Departmentof CSE and ICT Jaypee University of Information Technology, Waknaghat, Solan, H.P (INDIA)M. Young, TheTechnical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

2.”Ghazvinian, Holbert, Viswanathan. “SimpleWebScraping.”Internet:https://seanholbert.wordpress.com/2011/07/15/scrappy-simple-web scraping/, Jun.2015”

3.“Datahen."3 Advantages of web scrapingforyourenterprise"Internet:https://www.datahen.om/3-advantages-web-scraping enterprise/,May.17,2017”