

# HACKVOUCHER

## A PROJECT REPORT

Submitted to



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU

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

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

Dulla Pravalika	19BF1A0545
Gunda Sudarrshan	19BF1A0554
Puttur Nivas Kumar Reddy	19BF1A0569
Chakram Joycy Ikya	19BF1A0527

**Under the Esteemed Guidance of**

**Dr. K Santhi, Ph.D**

**Associate Professor**



*Department of Computer Science and Engineering*

**SRI VENKATESWARA COLLEGE OF ENGINEERING**

Permanent Affiliation to JNTUA & Approved by AICTE Recognized under Sections 2(f) & 12(B) of UGC act 1956. Accredited by NBA, New Delhi & NAAC Bangalore with 'A' Grade, Tirupati-517507, Chittoor, A.P.

2017-2021

# **SRI VENKATESWARA COLLEGE OF ENGINEERING**

Permanent Affiliation to JNTUA & Approved by AICTE Recognized under Sections 2(f) & 12(B) of UGC act 1956. Accredited by NBA, New Delhi & NAAC Bangalore with 'A' Grade, Tirupati-517507, Chittoor, A.P.

## **Department of Computer Science and Engineering**



### **CERTIFICATE**

This is to certify that the project entitled “**HACKVOUCHER**” is a bonafide work done by “**Dulla Pravalika 19BF1A0545, Gunda Sudarrshan 19BF1A0554, Puttur Nivas Kumar Reddy 19BF1A0569, Chakram Joycy Ikya 19BF1A0527**”, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering, from Jawaharlal Nehru Technological University Anantapur, Ananthapuramu during the year 2020-2021.

#### ***Project Guide***

**Dr. K Santhi**  
M.Tech, Ph.D(CSE)  
Department of CSE  
SV College of Engineering.  
Tirupati – 517 507

#### ***Head of the Department***

**Dr. K Santhi**  
M.Tech, Ph.D(CSE)  
Department of CSE  
SV College of Engineering.  
Tirupati – 517 507

**Viva-Voce date:** \_\_\_\_\_

**Internal Examiner**

**External Examiner**

## **DECLARATION**

We hereby declare that the project report entitled “**HACKVOUCHER**” done by us under the esteemed guidance of **Dr. K Santhi, Associate Professor**, and is submitted in partial fulfillment of the requirements for the award of the Bachelor’ s of Technology in **Computer Science and Engineering**.

Date:

Place:

**Dulla Pravalika (19BF1A0545)**

**Gunda Sudarrshan (19BF1A0554)**

**Puttur Nivas Kumar Reddy (19BF1A0569)**

**Chakram Joycy Ikya (19BF1A0527)**

## **ACKNOWLEDGEMENT**

We are thankful to our guide **Dr. K Santhi, M.Tech, Ph.D (CSE) and Head of The Department** for her valuable guidance and encouragement. Her helping attitude and suggestions have helped us in the successful completion of the project.

We are thankful to project coordinator **Mr. R. Venkataramana, Assistant Professor**, and **Mr. N. Narendra Reddy, Assistant Professor** for his guidance and regular schedules.

We would like to express our gratefulness and sincere thanks to **Dr. K Santhi, Associate Professor**, for her kind help and encouragement during the course of our study and in the successful completion of the project work.

We have great pleasure in expressing our hearty thanks to our beloved **Principal Dr. N. Sudhakar Reddy** for spending his valuable time with us to complete this project.

Successful completion of any project cannot be done without proper support and encouragement. We sincerely thank the **Management** for providing all the necessary facilities during the course of study.

We would like to thank our parents and friends, who have the greatest contributions in all our achievements, for the great care and blessings in making us successful in all our endeavors.

**Dulla Pravalika (19BF1A0545)**

**Gunda Sudarrshan (19BF1A0554)**

**Puttur Nivas Kumar Reddy (19BF1A0569)**

**Chakram Joycy Ikya (19BF1A0527)**

## **CONTENTS**

<b>CHAPTER NO.</b>	<b>DESCRIPTION</b>	<b>PAGENO.</b>
	Abstract	i
	List of Figures	ii
	List of Abbreviations	iii
<b>CHAPTER NO.</b>	<b>DESCRIPTION</b>	<b>PAGE NO.</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
<b>2</b>	<b>PROJECT DESCRIPTION</b>	<b>2</b>
	2.1. PROBLEM DEFINITION	2
	2.2. PROJECT DETAILS	3
<b>3</b>	<b>COMPUTATIONAL ENVIRONMENTS</b>	<b>5</b>
	3.1. SOFTWARE REQUIREMENTS	5
	3.2. HARDWARE REQUIREMENTS	5
	3.3. SOFTWARE FEATURES	5
<b>4</b>	<b>FEASIBILITY STUDY</b>	<b>24</b>
	4.1. TECHNICAL FEASIBILITY	24
	4.2 ECONOMICAL FEASIBILITY	24
	4.3 SOCIAL FEASIBILITY	24
<b>5</b>	<b>SYSTEM ANALYSIS</b>	<b>25</b>
	5.1. EXISTING SYSTEM	25
	5.1.1 DRAWBACKS	25
	5.2. PROPOSED SYSTEM	25
	5.2.1 FEATURES	25
	5.2.2 ADVANTAGES	26
<b>6</b>	<b>SYSTEM DESIGN</b>	<b>27</b>
	6.1. UML DIAGRAMS	27
<b>7</b>	<b>SYSTEM IMPLEMENTATION</b>	<b>32</b>
	7.1 ONBOARDING FLOW	32

	7.2 DASHBOARD	32
	7.3 PROFILE	36
8	ER DIAGRAM	37
9	SYSTEM TESTING	39
	9.1 UNIT TESTING	39
	9.2 INTEGRATION TESTING	40
	9.3 SYSTEM TESTING	41
	9.4 ACCEPTANCE TSETING	42
10	SAMPLE SOURCE CODE	43
11	RESULTS & SCREEN LAYOUTS	55
12	CONCLUSION	59
13	FUTURE ENHANCEMENTS	60
14	BIBLIOGRAPHY	61
	14.1 REFERENCES	61
	14.2 WEBSITES	61

## ABSTRACT

As a result of the pandemic, our country has undergone a wave of digital transformation over the past two years. As a result of lack of choice, concerns about safety, or just convenience, consumers have been taking to digital means to fulfill their needs. This change in behavior boosted the growth of the e-commerce industry in India. Due to an increase in internet and smartphone penetration, that segment had already experienced rapid growth, but the pandemic tripled it. Payments for everything, from groceries to clothes to books to personal care products, are now made online. This has led to the growth of e-commerce companies like Flipkart, Amazon, and Myntra, Zomato etc., that offer vouchers/gifts cards to promote the online transactions and payments. The voucher/gifts that we receive from e-commerce website is getting waste because most of them are not using, so by seeing this scenario we motivate to use these vouchers to exchange with other so that these vouchers are used by other and we also used other vouchers that are of our use by exchanging with money or point. So basically, we created a full stack web application, where users can buy and sell the voucher. Voucher are the e-commerce rewards or virtual gifts that users get when they do any kind of transaction from these e-commerce websites. E-commerce websites that provide vouchers are google pay, amazon pay, bhim upi.

The objective of this abstract is to develop a full stack web application system for buy and sell the unused vouchers. The system was developed using NodeJS and MongoDB as server-side programming and database respectively.

## LIST OF FIGURES

S. NO.	FIGURE NO.	DESCRIPTION	PAGE NO.
1	Fig 3.3.1.1	JDK Download	7
2	Fig 3.3.1.2	JDK Installation	7
3	Fig 3.3.1.3	Installation of JDK 8	8
4	Fig 3.3.1.4	Java SE Development Kit	8
5	Fig 3.3.1.5	Path Selection	10
6	Fig 3.3.1.6	Select Variable Value	11
7	Fig 3.3.1.7	Set Class Path	11
8	Fig 3.3.1.8	Successful Installation of JDK	12
9	Fig 3.3.1.9	Android Architecture	17
10	Fig 3.3.1.10	Select Preferences	22
11	Fig 3.3.1.11	Android SDK and AVD Manager	23
12	Fig 3.3.1.12	Installing Packages	24
13	Fig 3.3.1.13	Eclipse SDK	28
14	Fig 3.3.1.14	Plug-In Development	30
15	Fig 3.3.1.15	Android Virtual Devices	30
16	Fig 3.3.1.16	Create new Android Virtual Device	31
17	Fig 3.3.1.17	Testing Device	31
18	Fig 6.1.1	Use Case Diagram	36
19	Fig 6.1.2	Class Diagram	37
20	Fig 6.1.3	Sequence Diagram	37
21	Fig 6.1.4	Collaboration Diagram	38
22	Fig 6.1.5	Activity Diagram	39
23	Fig 6.1.6	Component Diagram	40
24	Fig 6.1.7	Deployment Diagram	40
25	Fig 7	ER Diagram	41
26	Fig 9	Software Development Life Cycle	44



## **LIST OF ABBREVIATIONS**

<b>S.NO</b>	<b>ACRONYM</b>	<b>ABBREVIATION</b>
1	GPS	Global Positioning System
2	JDK	Java Development Kit
3	IDE	Integrated Development Environment
4	SDK	Software Development Kit
5	AVD	Android Virtual Device
6	AOSP	Android Open-Source Project
7	ADT	Android Development Tools
8	DVM	Dalvik Virtual Machine
9	JVM	Java Virtual Machine
10	UML	Unified Modelling Language

## **CHAPTER 1 INTRODUCTION**

Hack Voucher is a full stack web application, where users can buy and sell the voucher. Voucher are the ecommerce rewards or virtual gifts that user gets when they do any kind of transaction from these ecommerce websites. E-commerce websites that provide vouchers are google pay, amazon pay, Bhim UPI.

In this project we are making voucher hosting platform in which we get some vouchers from different e-commerce website like google pay, amazon pay, bhim upi etc which are when we redeem, some come out blank and some of no use of us. If we have got the coupon, which is our waste coupon/that is of no use we can host these coupons on this website and the user who have need of that coupon buy in exchange of coins. So basically, In this Project which we are building, a voucher hosting platform user can sell vouchers there. and can purchase useful vouchers like discount coupons, cashback coupons etc.

This is the key of the project. We also get some bank discounts, coupons from shopping websites like myntra, flipkart, amazon. Some trading account coupons that are of no use for us, but can be useful to others. Our Platform will identify blank vouchers, or non-usable to us and we can use them for our profit to earn money. Here user can buy and sell their voucher at the same place without moving back and forth at a dashboard that is on a home where all the listed vouchers are hosted.

## **CHAPTER 2**

### **PROJECT DESCRIPTION**

#### **2.1 PROBLEM DEFINITION**

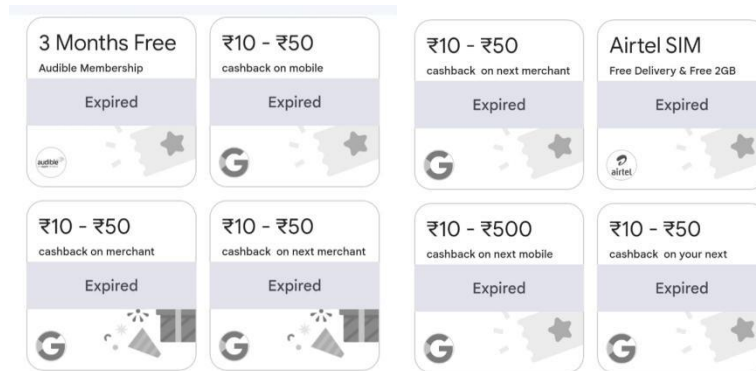
With the increasing popularity of digital transactions and online shopping in India, various online payment apps and e-commerce platforms have introduced various offers and discounts to attract more users and customers. These offers range from cashback rewards to discount coupons that can be redeemed on future purchases.

This trend has not only made online transactions more convenient for people but has also given them an incentive to adopt these digital payment methods. The availability of offers and discounts has also encouraged small businesses and merchants to accept digital payments, thereby promoting financial inclusion and reducing the dependence on cash transactions.

Furthermore, e-commerce platforms like Flipkart, Amazon, and Myntra use discount coupons to incentivize customers to make purchases from their platforms. These coupons offer discounts on products, free shipping, or even cashback rewards. This not only helps these platforms increase their market share but also provides customers with a cost-effective way to shop online.

Overall, the availability of offers and discounts has made digital transactions and online shopping more accessible and attractive to people in India, contributing to the growth of the digital economy in the country.

But many coupons and vouchers can go to waste due to individual sharing and lack of efficient use. While some online platforms allow users to share their coupons with friends and family, there is no centralized platform for multiple sharing of coupons. The coupons that are not useful for the customer are let alone and they get expired after some time. But those expired coupons may be useful for some other user.



**Fig 2.1 Coupons getting expired**

## 2.2 PROJECT DETAILS

Developing a platform where users can exchange vouchers and coupons was the proposed idea to reduce wastage and promote more efficient use of these offers. The platform can provide an interactive UI to make it easy for users to buy and sell coupons. Here are some features that the platform could include:

1. User registration and login: The platform should have a registration and login system to authenticate users and ensure the security of their information.
2. Search and filter: The platform should allow users to search for coupons based on various criteria such as category, discount value, validity period, etc.
3. Coupon details and images: The platform should display the details of the coupon such as the discount value, validity period, terms and conditions, and any other relevant information. The platform should also allow users to upload images of the coupon.
4. Coupon purchase: The platform should allow users to buy coupons using a payment gateway.
5. Coupon selling: The platform should allow users to sell their unused or unwanted coupons. The sellers can set a price for their coupons, and the platform can charge a commission on the sale.
6. Admin validation: After a seller submits a coupon, the admin can review the details and validate the coupon. The admin will check the coupon's validity period, terms and conditions, and other relevant details to ensure its authenticity. If the coupon is valid, the admin will approve it for sale on the platform.

## **HACK VOUCHER**

---

7. User profile: The platform should allow users to create and manage their profile, including their personal information, payment methods, and purchase/sales history.

8. Notification system: The platform should have a notification system to inform users of new coupons, purchases, and sales.

Overall, developing such a platform can be a great way to reduce the wastage of coupons and vouchers and promote their efficient use.

## CHAPTER 3

### COMPUTATIONAL ENVIRONMENT

#### 3.1 SOFTWARE SPECIFICATION

- Operating System - Windows 7/8/10
- IDE - Visual Studio Code.
- Libraries Used - Bootstrap, NPM Modules.
- Technologies - ExpressJs, Mongo DB.

#### 3.2 HARDWARE SPECIFICATION

- Processor - I3/Intel Processor
- RAM - 8 GB
- Hard Disk - 160 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA

#### 3.3 Software Features

Software Installations for a Full-Stack NodeJS project

##### 3.3.1 Software Installation of NodeJS

**Node.js** is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

##### 3.3.2 Procedure to Install Node.js on Windows

Since we have decided to create an application using Node.js, first of all, you have to install Node.js on your Windows system.

Here, we are going to explain the installation process step-by-step. So, let's start with the first step now.

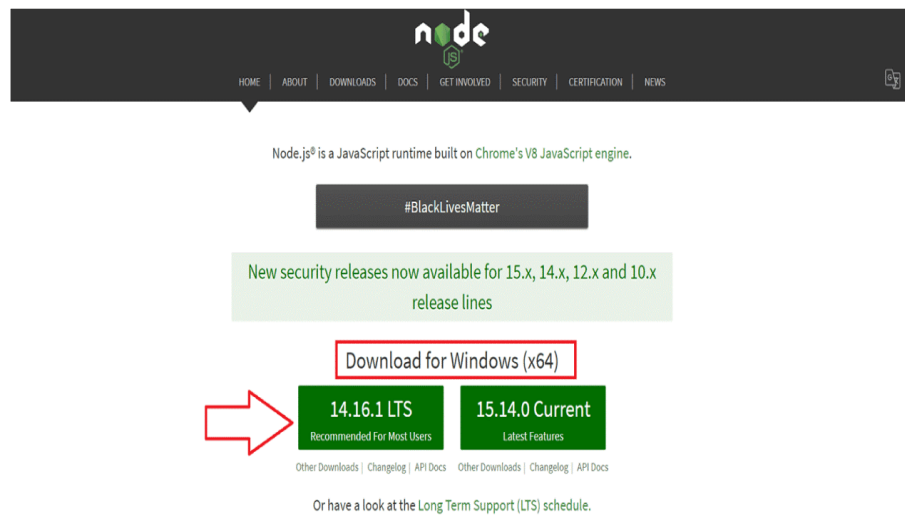
# HACK VOUCHER

---

## Step 1: Download the Installer

Download the **Windows Installer** from [NodeJs official website](https://nodejs.org/en/download/). Make sure you have downloaded the latest version of NodeJS. It includes the NPM package manager.

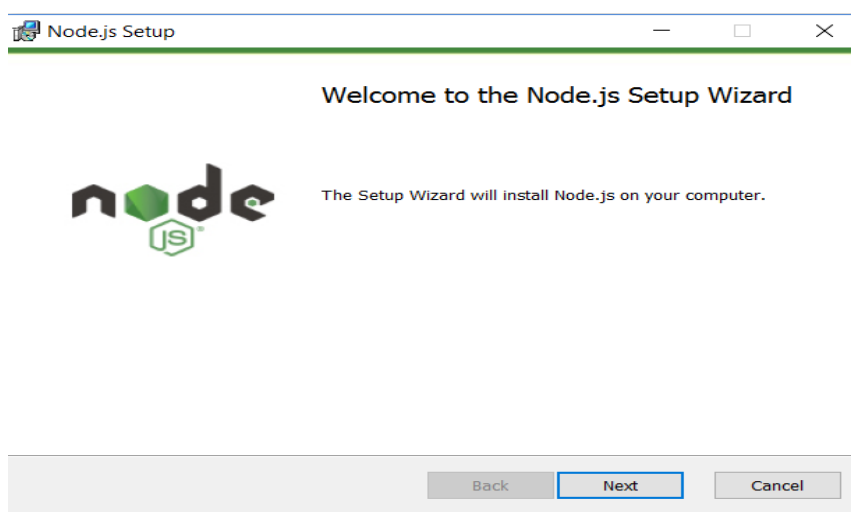
Here, we are choosing the 64-bit version of the Node.js installer.



The LTS (Long-term Support) version is highly recommended for you. After the download of the installer package, install it with a double-click on it.

Now .msi file will be downloaded to your browser. Choose the desired location for that.

## Step 2: Install Node.js and NPM



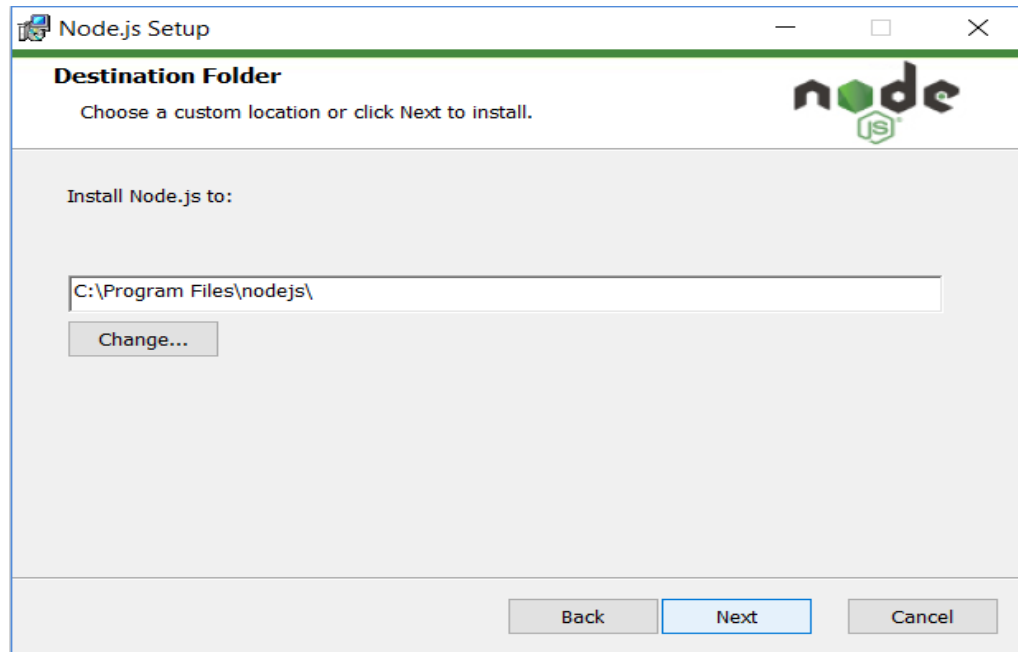
After choosing the path, double-click to install .msi binary files to initiate the installation process. Then give access to run the application.

## HACK VOUCHER

---

You will get a welcome message on your screen and click the “Next” button. The installation process will start.

- Choose the desired path where you want to install Node.js.



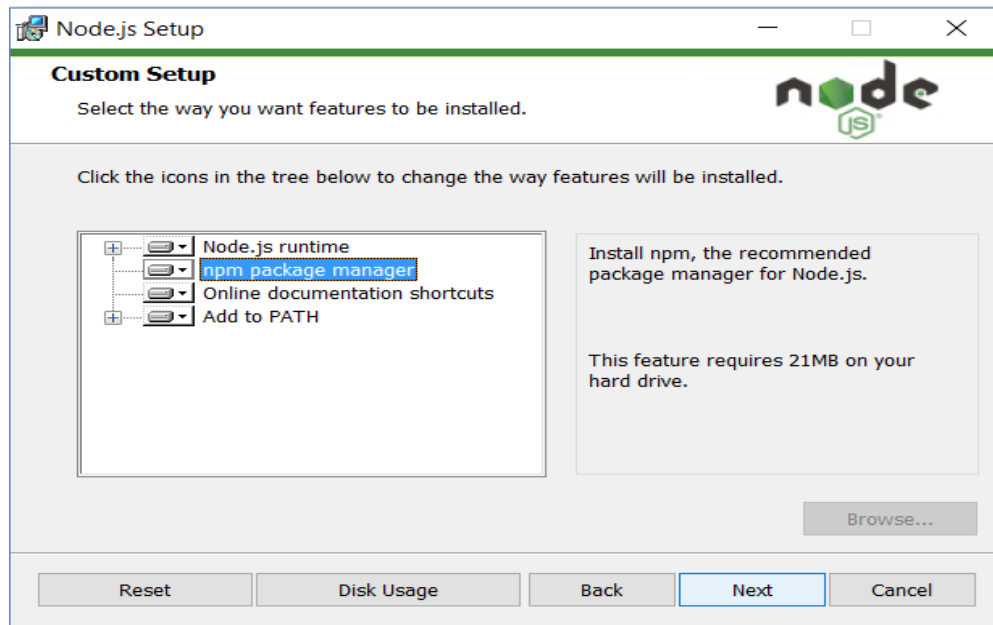
- By clicking on the Next button, you will get a custom page setup on the screen. Make sure you choose **npm package manager** , not the default of **Node.js runtime** . This way, we can install Node and NPM simultaneously.

You should have 143MB of space to install Node.js and npm features.

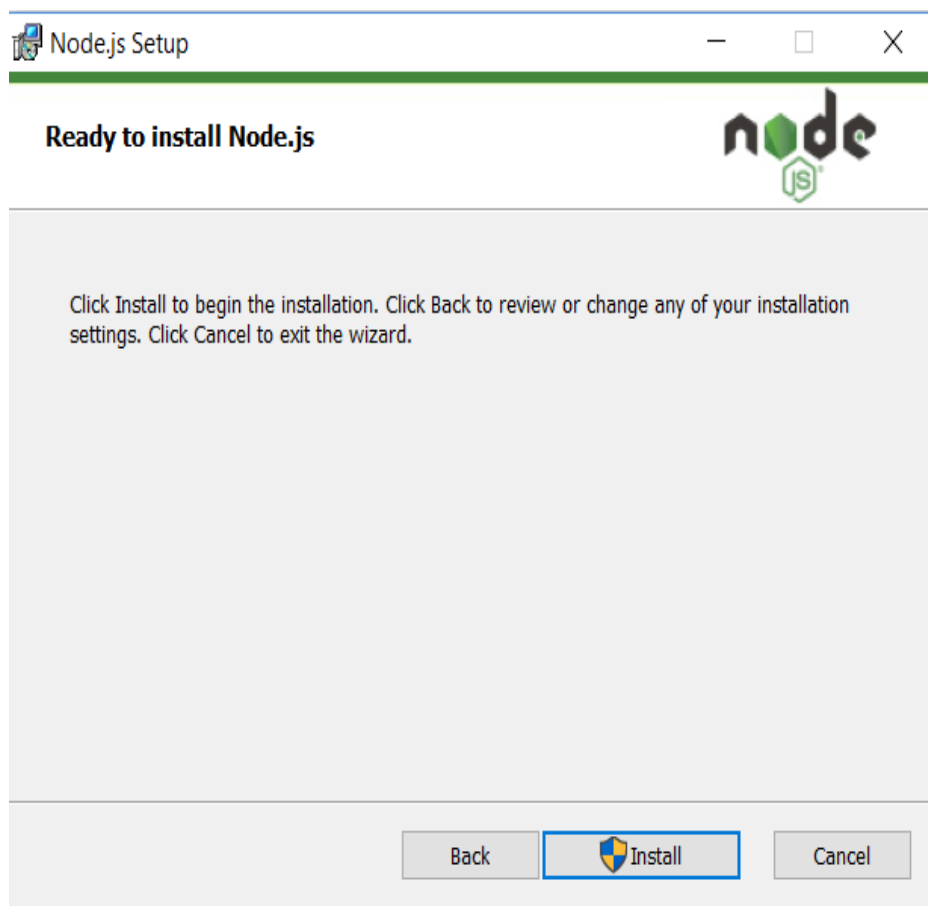
The following features will be installed by default:

- Node.js runtime
- Npm package manager
- Online documentation shortcuts
- Add to Path





- Bang! The setup is ready to install Node and NPM. Let's click on the **Install** button so hard!



Step 3: Check Node.js and NPM Version using command prompt

- If you have a doubt whether you have installed everything correctly or not, let's verify it with "Command Prompt".

## HACK VOUCHER

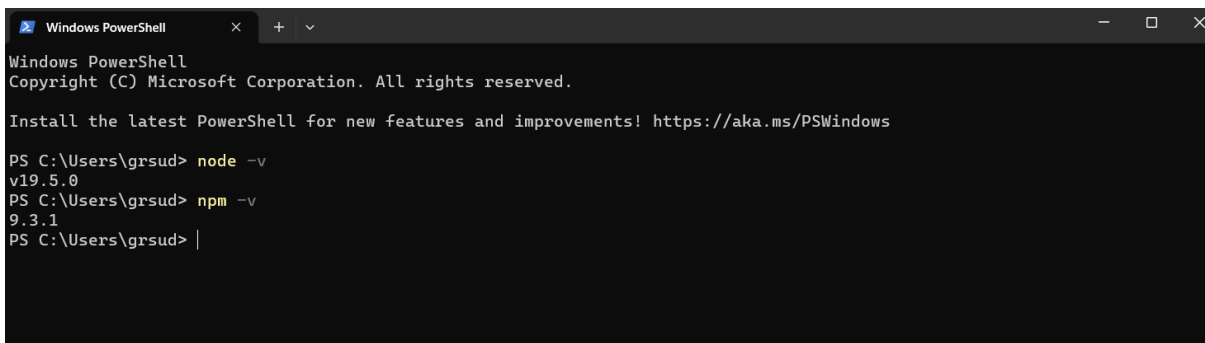
---

Command Prompt window will appear on the screen.

To confirm Node installation, type `node -v` command.

To confirm NPM installation, type `npm -v` command.

And you don't need to worry if you see different numbers than mine as Node and NPM are updated frequently

A screenshot of a Windows PowerShell terminal window. The title bar says 'Windows PowerShell'. The text inside shows the copyright notice for Microsoft Corporation and a link to install the latest PowerShell. Below that, the user runs 'node -v' and gets 'v19.5.0'. Then they run 'npm -v' and get '9.3.1'. The prompt is 'PS C:\Users\grsud> |'.

In my

case, the version of node.js is v19.5.0 and npm is 9.3.1.

### Adding to the path:

You should not need to do anything to the system variables, as the windows installer takes care of the system variables itself while installing through the .msi installer.

If you use any other format for installing node.js on your PC, you should put the system variable path for node.js as follows:

PATH: `C:\Users\{username}\AppData\Roaming\npm C:\Program Files\{path to the nodejs folder}`

for example:

PATH: `C:\Users\admin\AppData\Roaming\npm C:\Program Files\nodejs`

### Step 4: Updating the Local npm version.

The final step in node.js installed is the updation of your local npm version (if required) – the package manager that comes bundled with Node.js.

You can run the following command, to quickly update the npm

`npm install npm -global // Updates the 'CLI' client`

## 3.3.2 Procedure to install Visual Studio Code(VS Code) IDE

Visual Studio Code is the most popular code editor and the IDEs provided by Microsoft for writing different programs and languages. It allows the users to develop new code bases for

## HACK VOUCHER

---

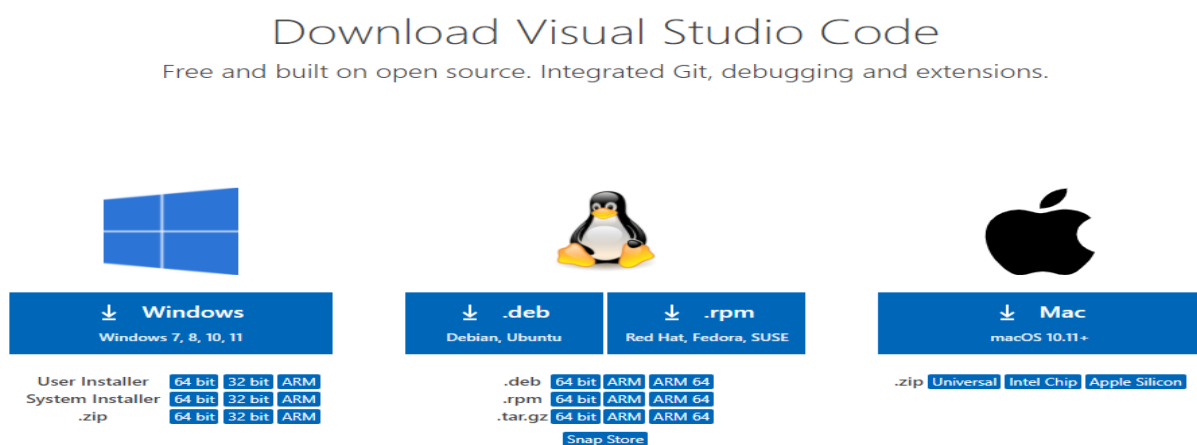
their applications and allow them to successfully optimize them and debug them properly. It is a very user-friendly code editor and it is supported on all the different types of operating systems like Windows, macOS, and Linux. It has the support for all the languages like C, C++, Java, Python, JavaScript, React, Node JS, etc.

It is the most popular code editor in India also. It allows users with different types of in-app installed extensions for the different types of their supported languages. It allows the programmers to write the code with ease with the help of these extensions. Also, Visual Studio Code has a great vibrant software UI with amazing night mode features. It suggests auto-complete code to the users which suggests the users complete their code with full ease.

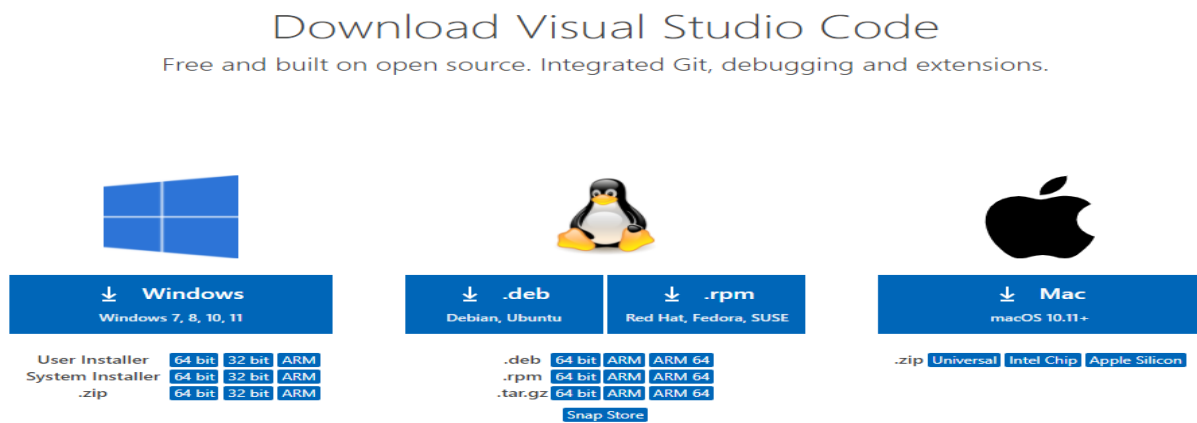
### Installing Visual Studio Code on Windows

Follow the below steps to install **Visual Studio Code** on Windows:

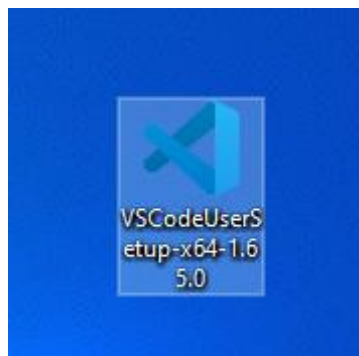
**Step 1:** Visit the [official website](#) of the **Visual Studio Code** using any web browser like Google Chrome, Microsoft Edge, etc.



**Step 2:** Press the “**Download for Windows**” button on the website to start the download of the Visual Studio Code Application.

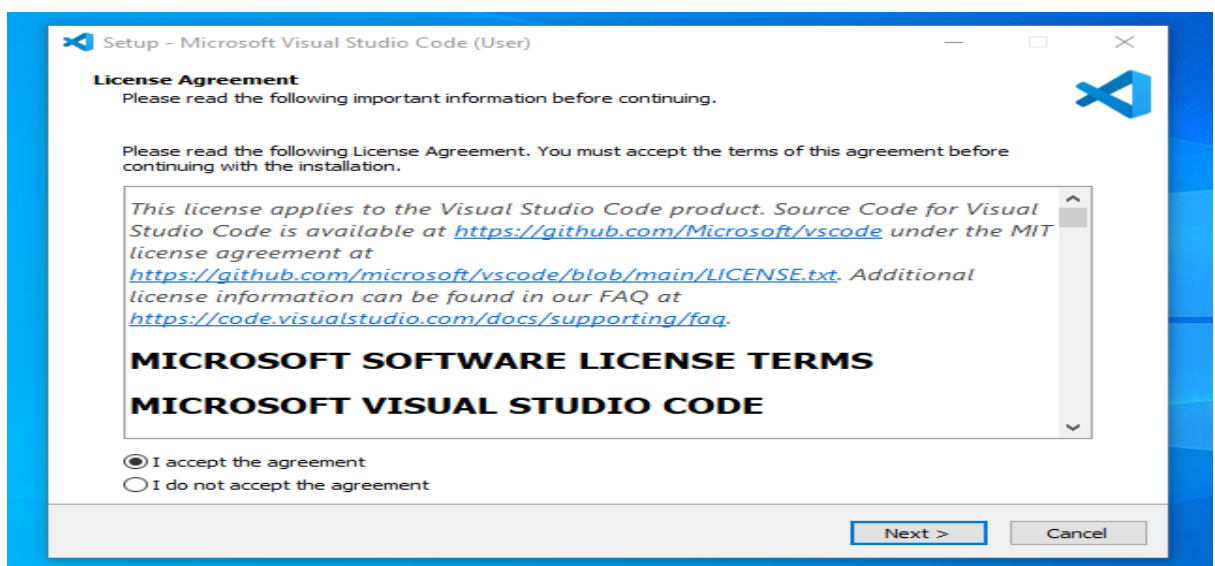


**Step 3:** When the download finishes, then the Visual Studio Code icon appears in the downloads folder.

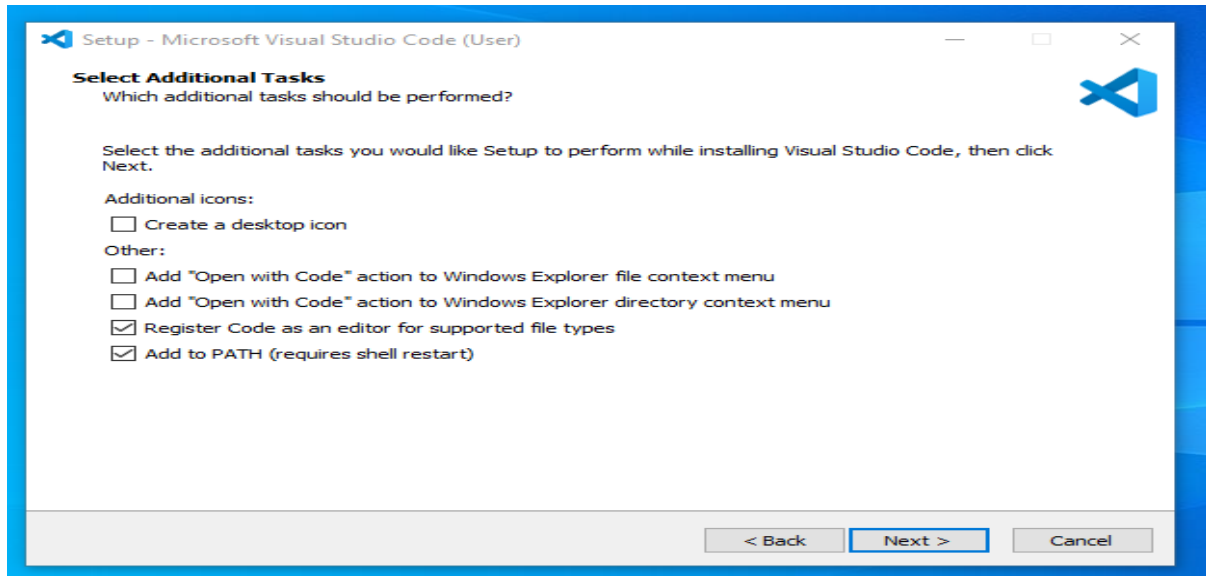


**Step 4:** Click on the installer icon to start the installation process of the Visual Studio Code.

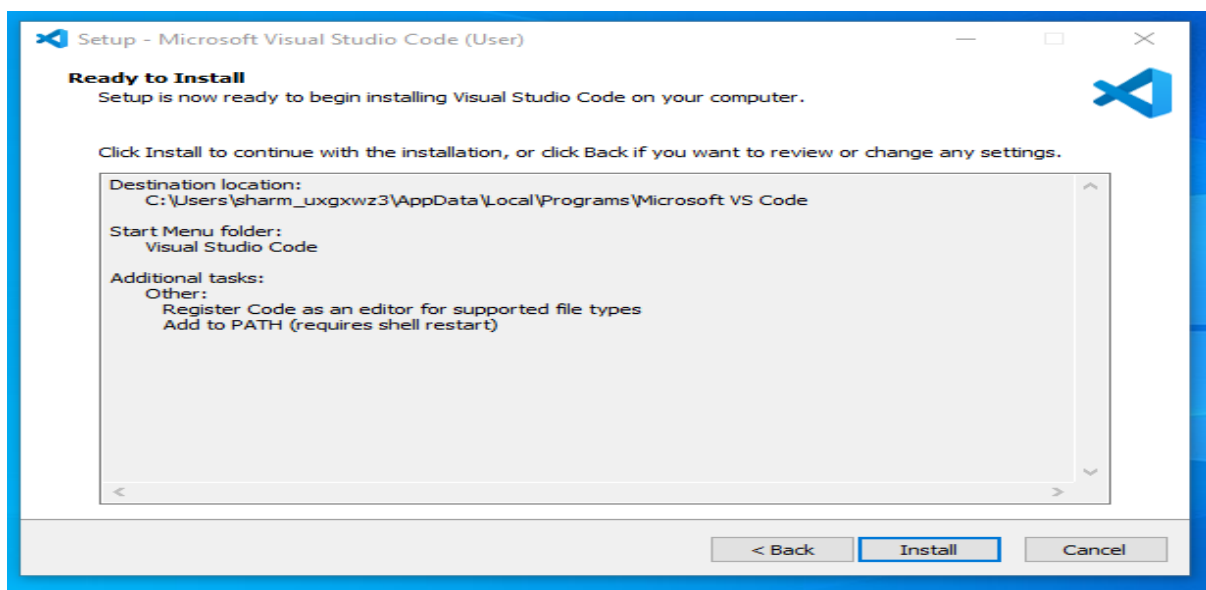
**Step 5:** After the Installer opens, it will ask you for accepting the terms and conditions of the Visual Studio Code. Click on **I accept the agreement** and then click the **Next** button.



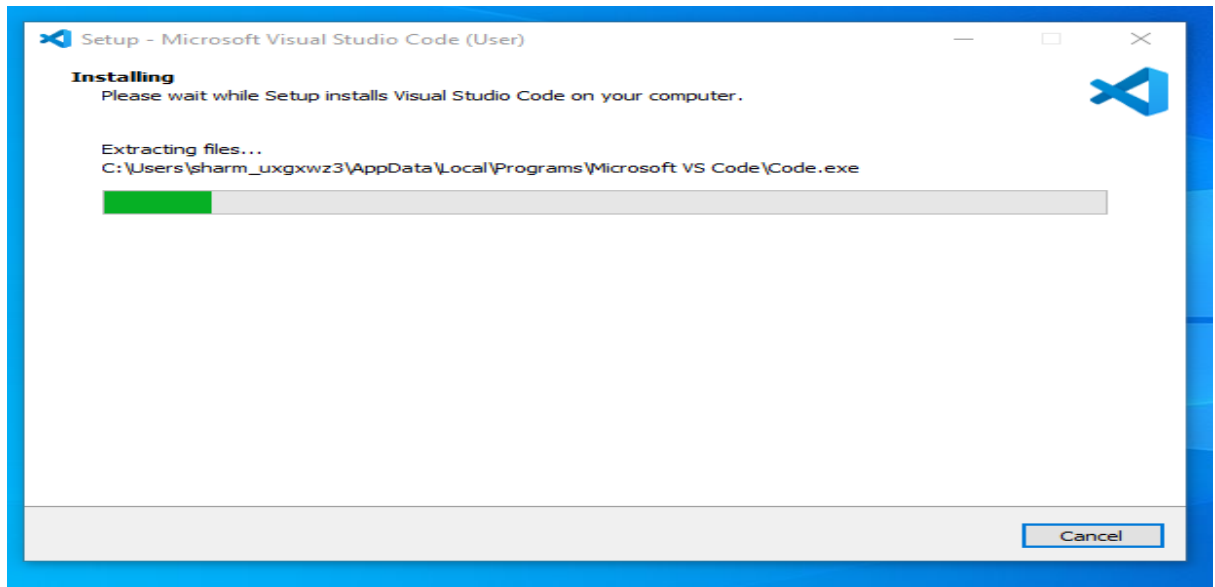
**Step 6:** Choose the location data for running the Visual Studio Code. It will then ask you for browsing the location. Then click on **Next** button.



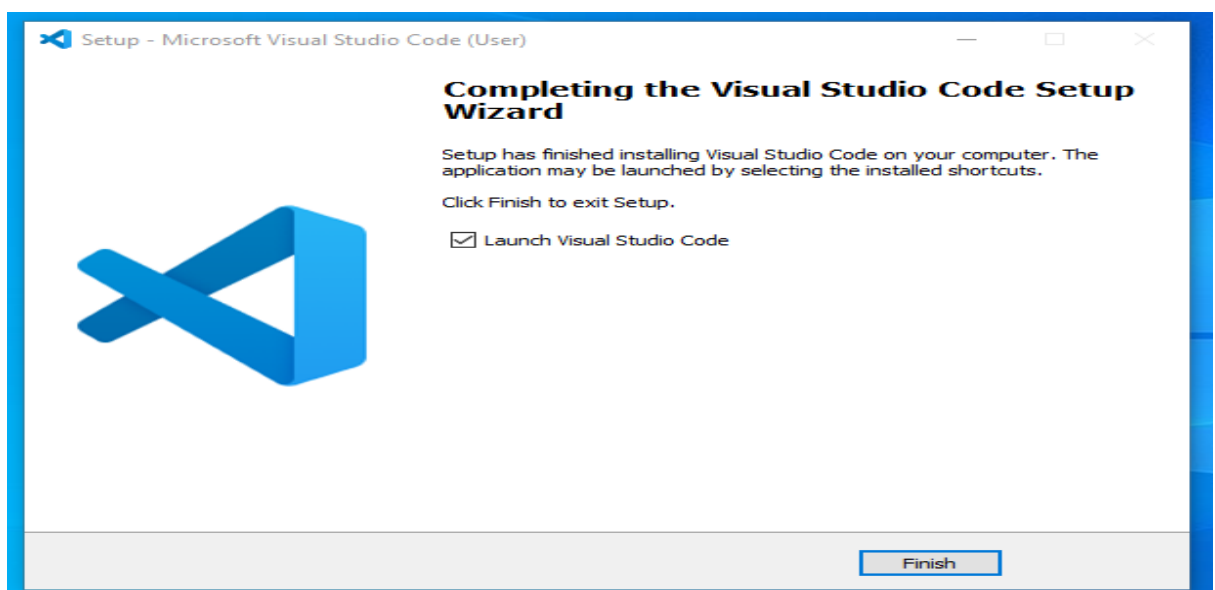
**Step 7:** Then it will ask for beginning the installing setup. Click on the **Install** button.



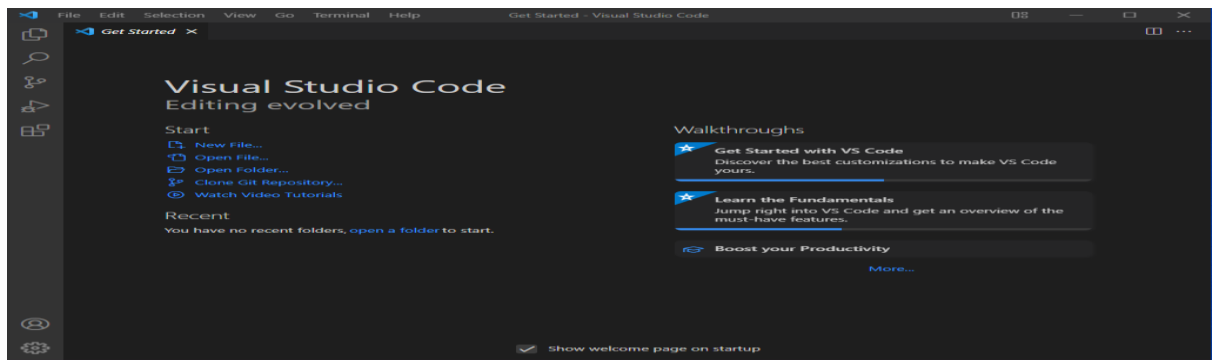
**Step 8:** After clicking on Install, it will take about 1 minute to install the Visual Studio Code on your device.



**Step 9:** After the Installation setup for Visual Studio Code is finished, it will show a window like this below. Tick the “**Launch Visual Studio Code**” checkbox and then click **Next**.



**Step 10:** After the previous step, the **Visual Studio Code window** opens successfully. Now you can create a new file in the Visual Studio Code window and choose a language of yours to begin your programming journey!

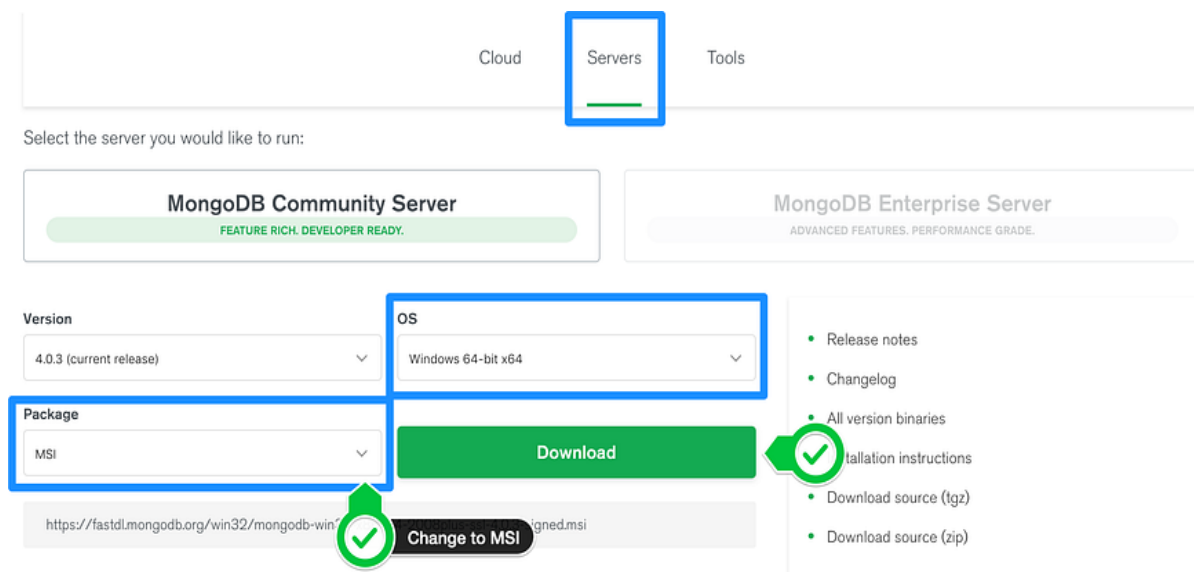


So this is how we successfully installed **Visual Studio Code** on our Windows system.

## Procedure to Install MongoDB

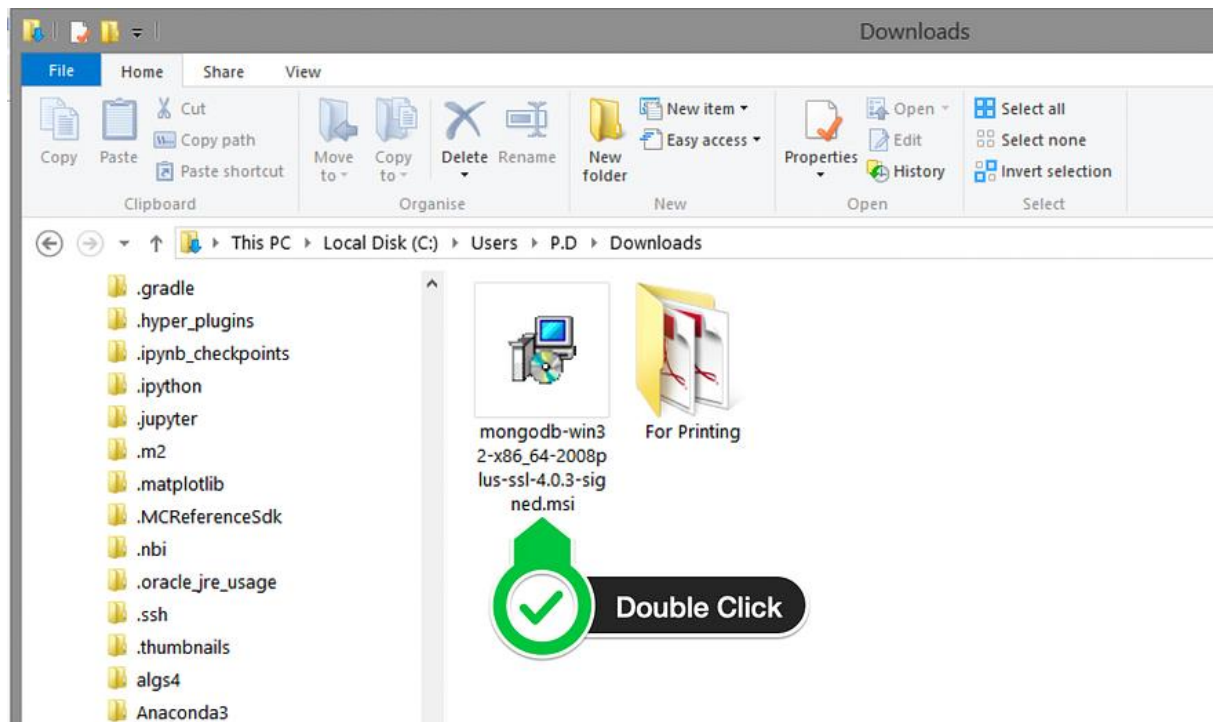
### Step 1 — Download the MongoDB MSI Installer Package

Head over [here](#) and download the current version of MongoDB. Make sure you **select MSI** as the package you want to download.



### Step 2 — Install MongoDB with the Installation Wizard

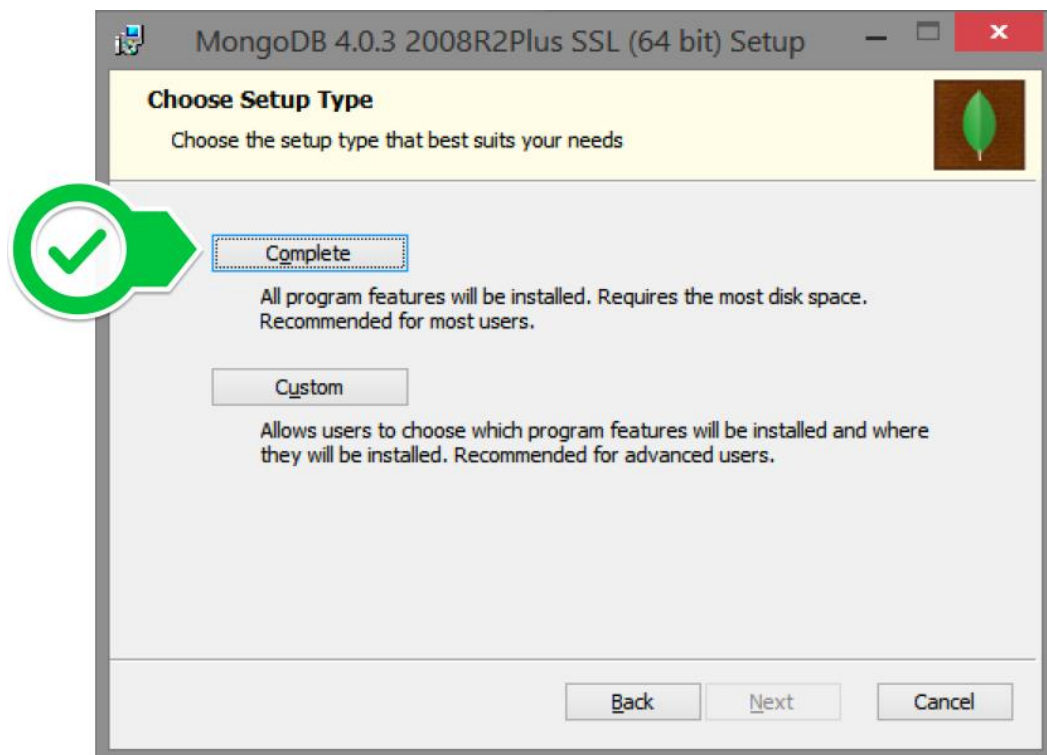
A. Make sure you are logged in as a user with Admin privileges. Then navigate to your downloads folder and double click on the .msi package you just downloaded. This will launch the installation wizard.



B. Click Next to start installation.

C. Accept the licence agreement then click Next.

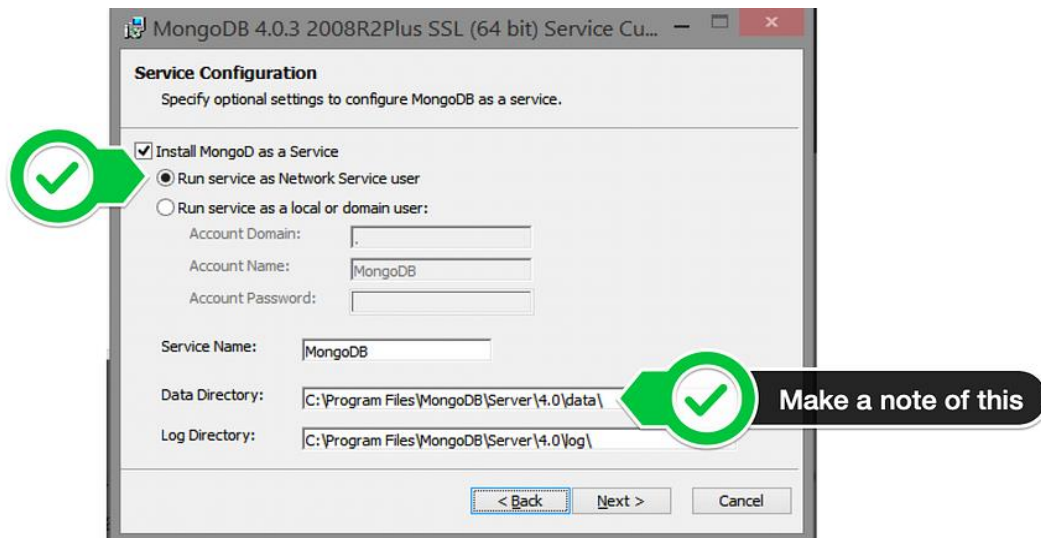
D. Select the Complete setup.



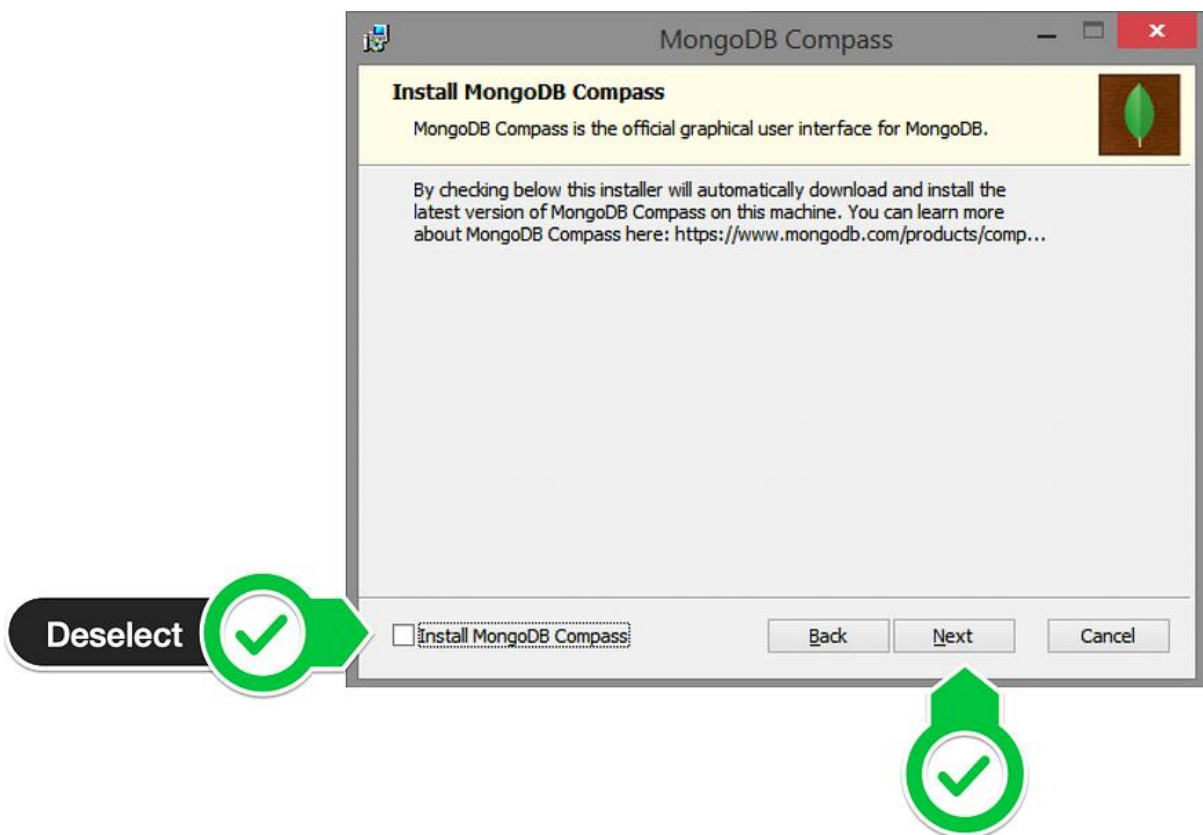


## HACK VOUCHER

E. Select “Run service as Network Service user” and make a note of the data directory, we’ll need this later.



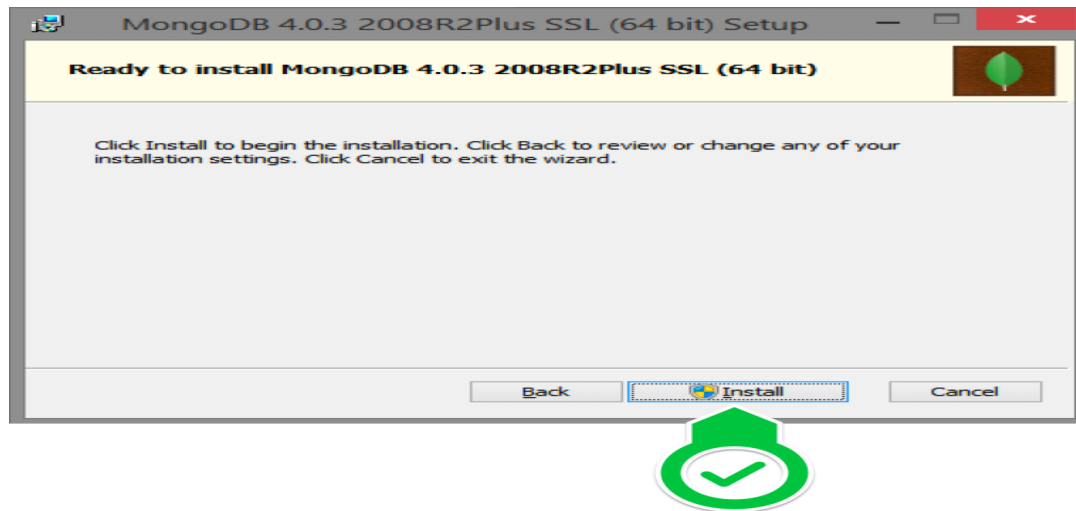
F. We won’t need Mongo Compass, so deselect it and click Next.



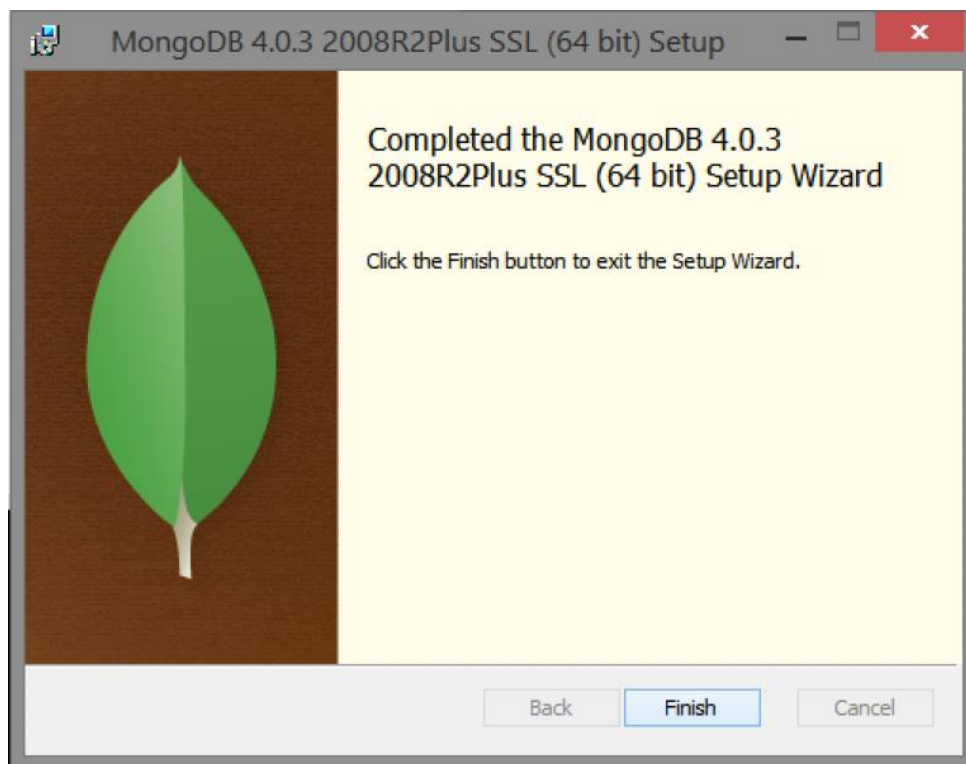
## HACK VOUCHER

---

G. Click Install to begin installation.

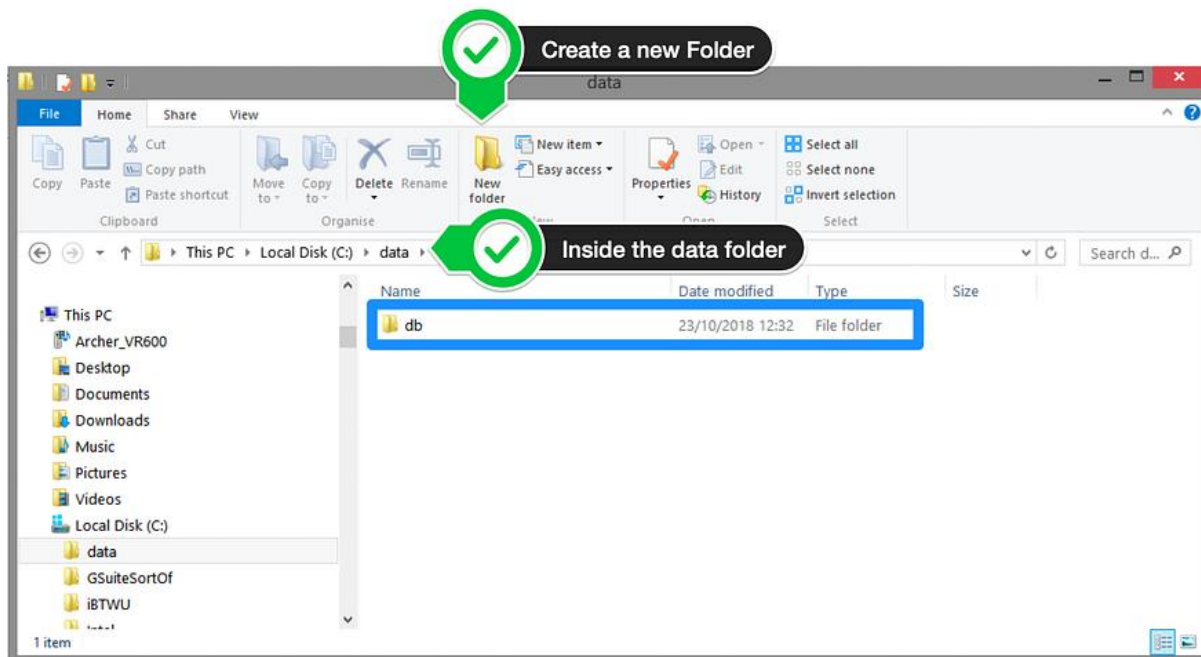


F. Hit Finish to complete installation.



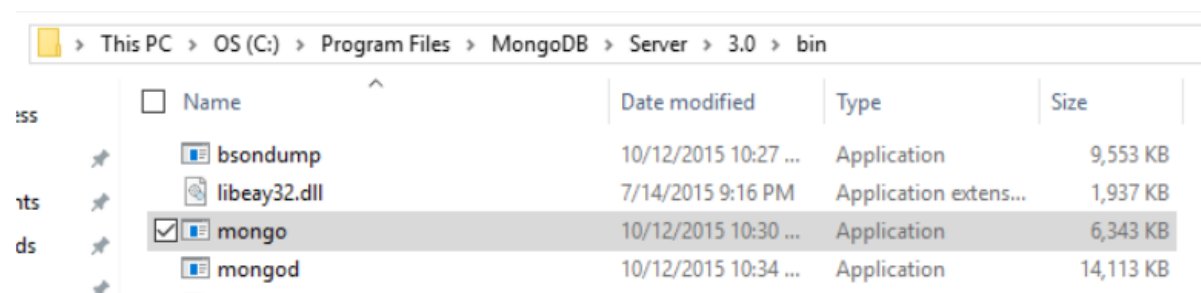
### Step 3— Create the Data Folders to Store our Databases

- A. Navigate to the **C Drive** on your computer using Explorer and create a new folder called **data**
- B. Inside the **data** folder you just created, create another folder called **db**.

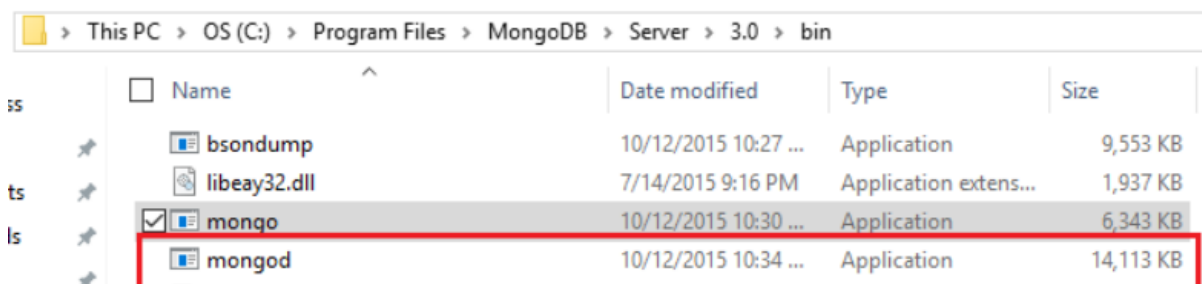


C. Add below path in the environment variable

D. Navigate to C:> ProgramFiles>MongoDB>Server>3.0>bin



E. Right click and run as administrator: Start the mongo db server



Once you start the server you will see a command prompt saying (1 connection now open)

### VS Code Features

1. **IntelliSense:** VS Code has built-in IntelliSense that provides intelligent code completion, suggesting variables, functions, and methods as you type, making it easier to write clean and error-free code.
2. **Debugging:** VS Code has powerful debugging tools that can help you identify and fix issues in your code. You can set breakpoints, step through your code, and inspect variables and data structures to identify and resolve issues quickly.
3. **Extensions:** VS Code has a vast library of extensions that can enhance your development experience. There are extensions for everything from code formatting to database management, making it easier to develop and deploy your MERN stack project.
4. **Git Integration:** VS Code has excellent Git integration that can make it easier to manage version control for your project. You can commit, push, pull, and merge changes right from the VS Code interface, making it easier to collaborate with other developers and track changes to your code.
5. **Live Server:** The Live Server extension allows you to see your changes in real-time as you edit your code. This makes it easier to test your MERN stack project and see the changes you make in real-time, improving your development workflow.
6. **Terminal Integration:** VS Code has a built-in terminal that you can use to execute commands and run scripts directly from the editor. This makes it easier to manage your project and perform tasks like installing dependencies or running tests.
7. **Code Formatting:** VS Code has built-in support for code formatting that can help you maintain consistent code style throughout your project. You can customize your code formatting rules to fit your project's needs, ensuring that your code is easy to read and maintain.

### MongoDB Features

MongoDB is a popular NoSQL database that has gained popularity for its flexibility, scalability, and ease of use. Unlike traditional SQL databases, which store data in tables and rows, MongoDB stores data in flexible, JSON-like documents that can vary in structure and schema. This allows developers to easily adapt to changing data models and build more agile and scalable applications.

In addition to its flexibility and scalability, MongoDB is also known for its performance and reliability. With built-in support for replication and automatic failover, MongoDB can provide high availability and fault tolerance, even in large-scale, distributed environments. Furthermore, its

## HACK VOUCHER

---

powerful querying and aggregation capabilities make it a popular choice for building applications that require complex data processing and analysis.

Overall, MongoDB's combination of flexibility, scalability, and performance make it a popular choice for building full-stack web applications. Whether you're building a small application or a large, distributed system, MongoDB can provide the flexibility, reliability, and performance you need to build a successful project.

1. **Scalability:** MongoDB is designed to scale horizontally, which means it can handle large amounts of data and traffic. This makes it an excellent choice for building full-stack web applications that require scalability.
2. **Flexibility:** MongoDB is a document-oriented database, which means it can store data in a flexible schema-less format. This makes it easy to adapt to changes in your application's data model and reduces the need for complex migrations.
3. **Querying:** MongoDB has a powerful query language that allows you to retrieve and manipulate data in a flexible and efficient way. This can help improve the performance of your full-stack web application and make it easier to work with data.
4. **Indexing:** MongoDB supports various types of indexing, including text indexing and geospatial indexing, which can help improve query performance and enable advanced data analysis.
5. **Replication:** MongoDB supports replica sets, which can help improve the availability and reliability of your full-stack web application. Replica sets provide automatic failover and can be used to distribute data across multiple servers for improved performance.
6. **Aggregation:** MongoDB has a powerful aggregation framework that allows you to perform complex data analysis and transformation operations. This can be useful for building full-stack web applications that require advanced data processing and analysis.
7. **Document validation:** MongoDB allows you to define document validation rules, which can help ensure data consistency and reduce the risk of data corruption. This can be useful for building full-stack web applications that require high data integrity and security.

### Modules and Libraries

Libraries and modules are essential tools for building a Full-Stack Web Application like HackVocher. They provide pre-built functionality for common tasks, such as handling HTTP requests, authenticating users, and processing payments. This saves time and effort, as

developers can focus on building the unique features of the application rather than reinventing the wheel.

Using libraries and modules also promotes code reusability and maintainability. Developers can use pre-built components in different parts of the application, reducing the amount of duplicated code and improving code organization. Furthermore, popular libraries and modules have large communities of developers contributing to their development and maintenance, ensuring that they are up-to-date and secure.

### Modules and Libraries that are used to develop HackVoucher

Here is a summary of how each of the dependencies listed can be used for a Full-Stack Web Application:

Module / Library Name: Version - description about the module

1. "connect-flash": "^0.1.1": Connect-flash is a middleware for Node.js that provides a way to store and retrieve flash messages in the session. Flash messages are messages that are displayed to the user for a short period, such as after a successful login or after an error occurs. Connect-flash is commonly used with Passport, a popular authentication library for Node.js, to display flash messages after authentication.
2. "connect-mongo": "^4.6.0": Connect-mongo is a middleware for Node.js that provides a way to store session data in MongoDB. It is commonly used with Express-session, a popular session management library for Node.js, to store session data in MongoDB rather than in memory. This makes it possible to share sessions across multiple servers in a load-balanced environment.
3. "cookie-parser": "^1.4.5": Cookie-parser is a middleware for Node.js that provides a way to parse cookies in HTTP requests. It is commonly used with Express, a popular web framework for Node.js, to parse cookies in requests and make them available in the request object.
4. "ejs": "^3.1.6": EJS is a popular templating language for Node.js that provides an easy way to generate HTML markup from data. It can be used to generate dynamic web pages by inserting data into pre-defined templates. EJS is commonly used with Express to generate HTML pages from server-side data.

5. "express": "^4.17.1": Express is a popular web framework for Node.js that provides a set of tools for building web applications. It provides a way to handle HTTP requests, define routes, and manage middleware. Express is commonly used to build RESTful APIs and web applications.
6. "express-session": "^1.17.2": Express-session is a popular library for managing sessions in Node.js. It provides a way to store session data on the server-side and make it available to the client-side. Express-session is commonly used with Connect-mongo to store session data in MongoDB.
7. "mongoose": "^6.0.10": Mongoose is a popular Object-Document Mapping (ODM) library for MongoDB. It provides an easy way to interact with MongoDB by defining schemas and models for data. Mongoose is commonly used with Express to store and retrieve data from MongoDB.
8. "morgan": "^1.10.0": Morgan is a middleware for Node.js that provides a way to log HTTP requests. It can be used to log requests to the console or to a file. Morgan is commonly used to monitor the activity of an application during development or production.
9. "multer": "^1.4.4": Multer is a middleware for handling file uploads in Node.js. It provides a way to handle multipart/form-data requests, allowing users to upload images, documents, and other files to the server.
10. "nodemon": "^2.0.13": Nodemon is a tool for automatically restarting a Node.js application when changes are detected in the source code. It is commonly used during development to automate the restart process and save time.
11. "passport": "^0.6.0": Passport is a popular authentication library for Node.js. It provides a way to authenticate users using a variety of strategies, including local authentication, OAuth, and OpenID. Passport is commonly used with Connect-flash to display flash messages after authentication.
12. "passport-local": "^1.0.0": Passport-local is a strategy for Passport that provides a way to authenticate users using a username and password. It is commonly used with Express and Mongoose to authenticate users.

## HACK VOUCHER

---

### Steps For Executing the Project

#### Step-1:

open Vs Code editor, enter into Project folder and click on terminal (available in top navigation bar of vs code editor) and enter into it.

#### Step-2:

Once you are in terminal Install necessary dependencies by running the command `npm install` or `npm i`  
This will install all the dependencies that are mentioned in the package.json file.

#### Step -3:

Now enter the command `npm run start` or `npm start` to initialise the project  
This will start the MongoDB database connection and will initiate local server on port 3000

#### Step-4:

Go to any browser and open `localhost::/3000/home` , you can access the project in your localhost web browser



### CHAPTER 4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY
- ECONOMICAL FEASIBILITY

#### 4.1 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. The developed system must have a modest requirement, as only minimal or no changes are required for implementing this system.

#### 4.2 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it, which is welcomed, as he is the final user of the system.

#### 4.3 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available.

## UNIT 5

### SYSTEM ANALYSIS

#### 5.1 EXISTING SYSTEM

Payments for everything, from groceries to clothes to books to personal care products, are now made online. This has led to the growth of e-commerce companies like Flipkart, Amazon, and Myntra, Zomato etc., that offer vouchers/gifts cards to promote the online transactions and payments. It is an online market place where user sell and buy the voucher that they get from different e-commerce website like google pay, amazon pay, Bhim UPI etc. and are of no use for a particular user but useful for other. Because of that many vouchers are getting wasted and are unused. So to overcome this problem in the wave of digital transformation we are building a Voucher hosting platform.

##### 5.1.1 DRAWBACKS OF EXISTING SYSTEM

- Wastage of coupons.
- Individual Sharing.
- No Alters regarding the coupons Deadline.

#### 5.2 PROPOSED SYSTEM:

- We can also develop interactive Web Apps and make it hosted in an public environment. Website provides more interactive UI and easy to use for users.
- We can also implement Filter in such a way that user view voucher of their interests and also get a notification on their profile and email when someone added a voucher of their interest.
- We will include Verification of Vouchers.
- User Profile Building (Based on Contribution).

##### 5.2.1 FEATURES OF PROPOSED SYSTEM:

- Register and Login functionality of Users.
- All available Vouchers can be seen.
- Buy and Update Vouchers.

##### 5.2.2 ADVANTAGES OF PROPOSED SYSTEM:

## **HACK VOUCHER**

---

- Better Usage of Coupons.
- Multiple sharing of Coupons.
- Efficient Alert system for Existing Coupons.
- Having competitive User Profile with contributions mechanism.

### CHAPTER 6 SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development, then design is the act of taking the marketing information and creating the design of the product to be manufactured.

Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

#### 6.1 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

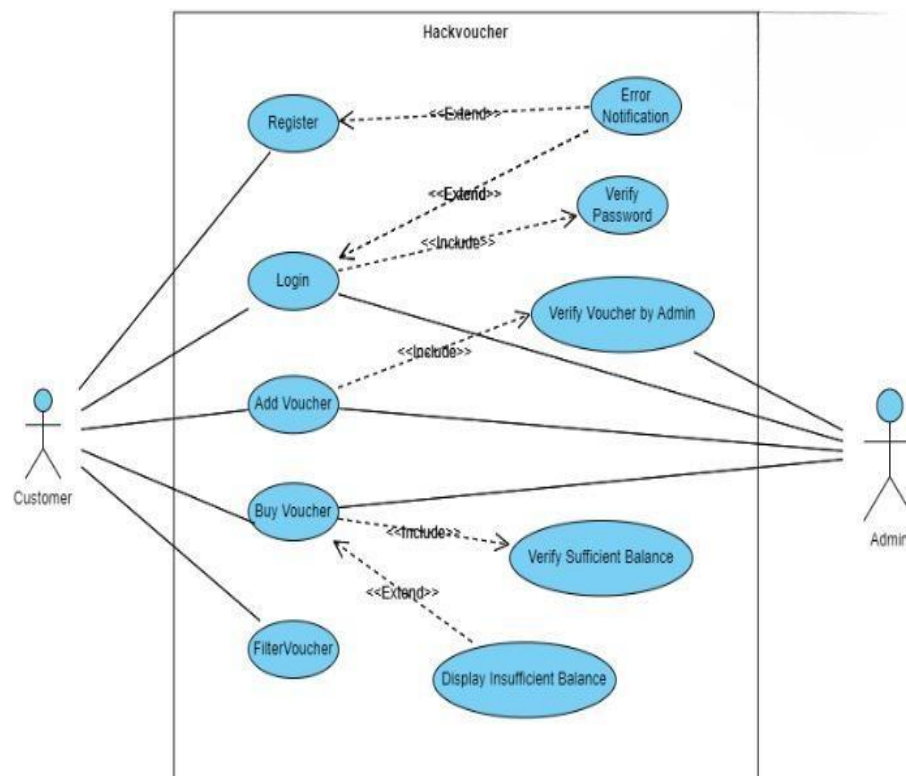
The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: A Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

### **6.1.1 USE CASE DIAGRAM:**

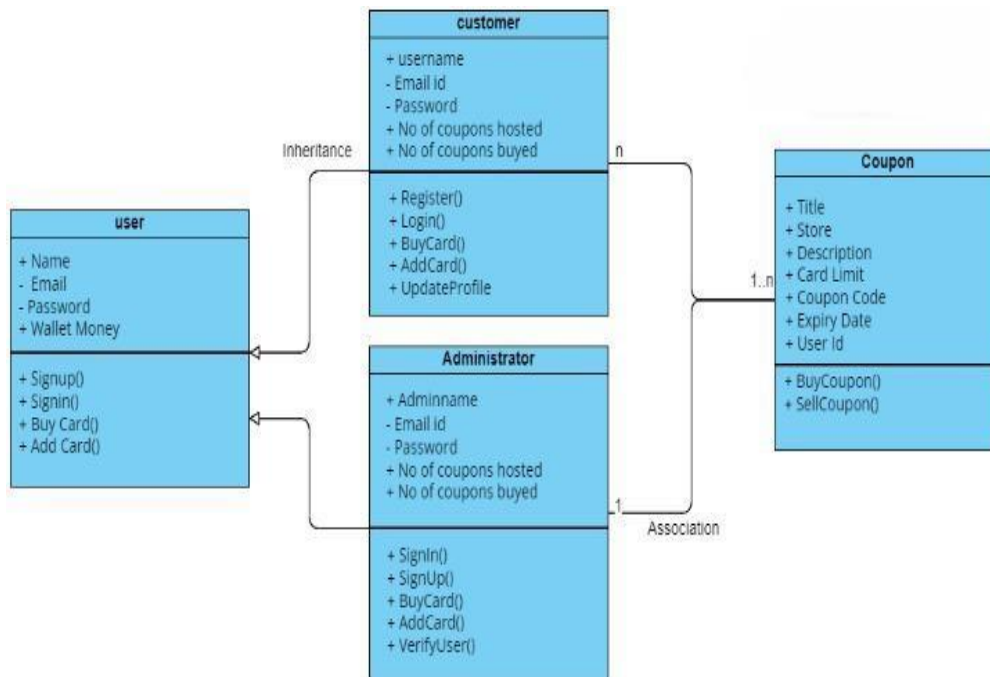
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



**Fig. 6.1.1: Use Case Diagram**

### **6.1.2 CLASS DIAGRAM:**

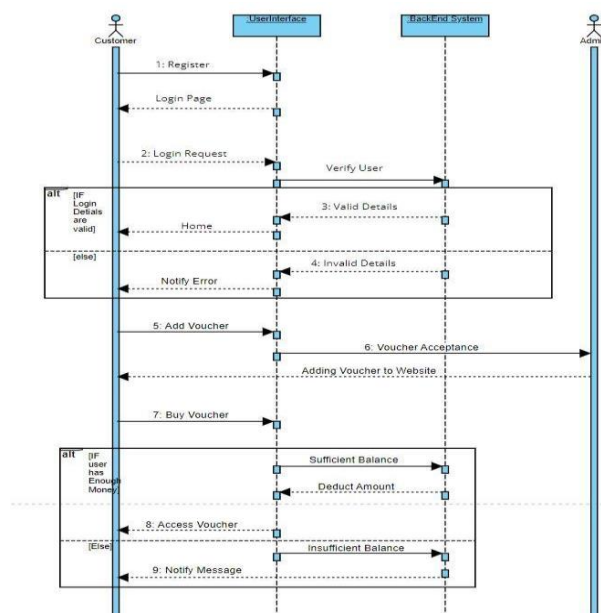
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**Fig.6.1.2: Class Diagram**

## 6.1.3 SEQUENCE DIAGRAM:

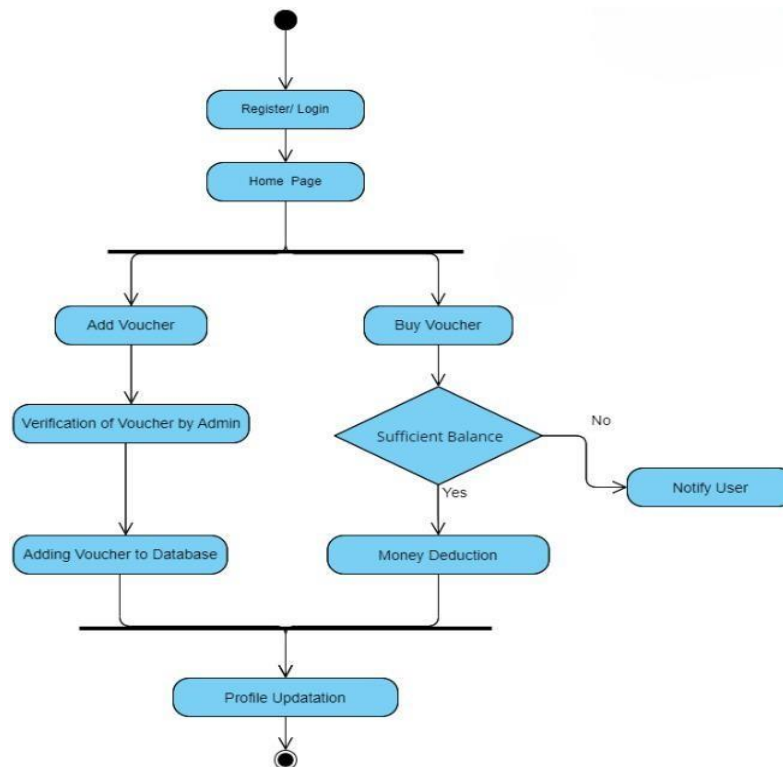
A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**Fig. 6.1.3: Sequence Diagram**

### **6.1.4 ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**Fig. 6.1.4: Activity Diagram**

### **6.1.5 COMPONENT DIAGRAM:**

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node. The purpose of a component diagram is to show the relationship between different components in a system.

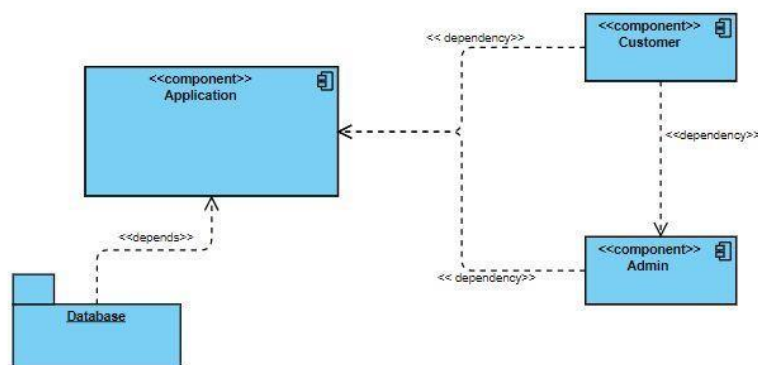


Fig. 6.1.5: Component Diagram

## 6.1.6 DEPLOYMENT DIAGRAM:

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

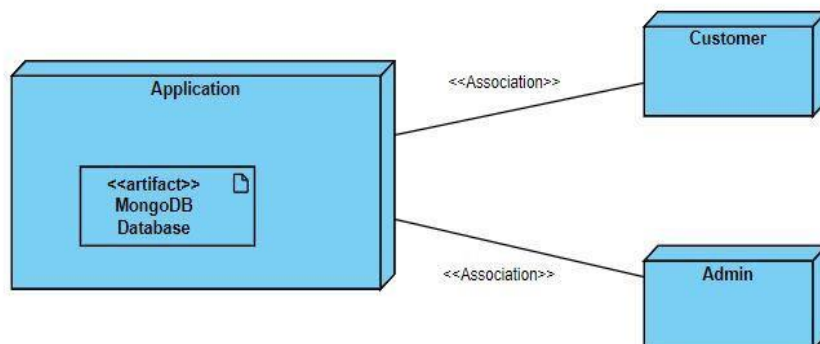


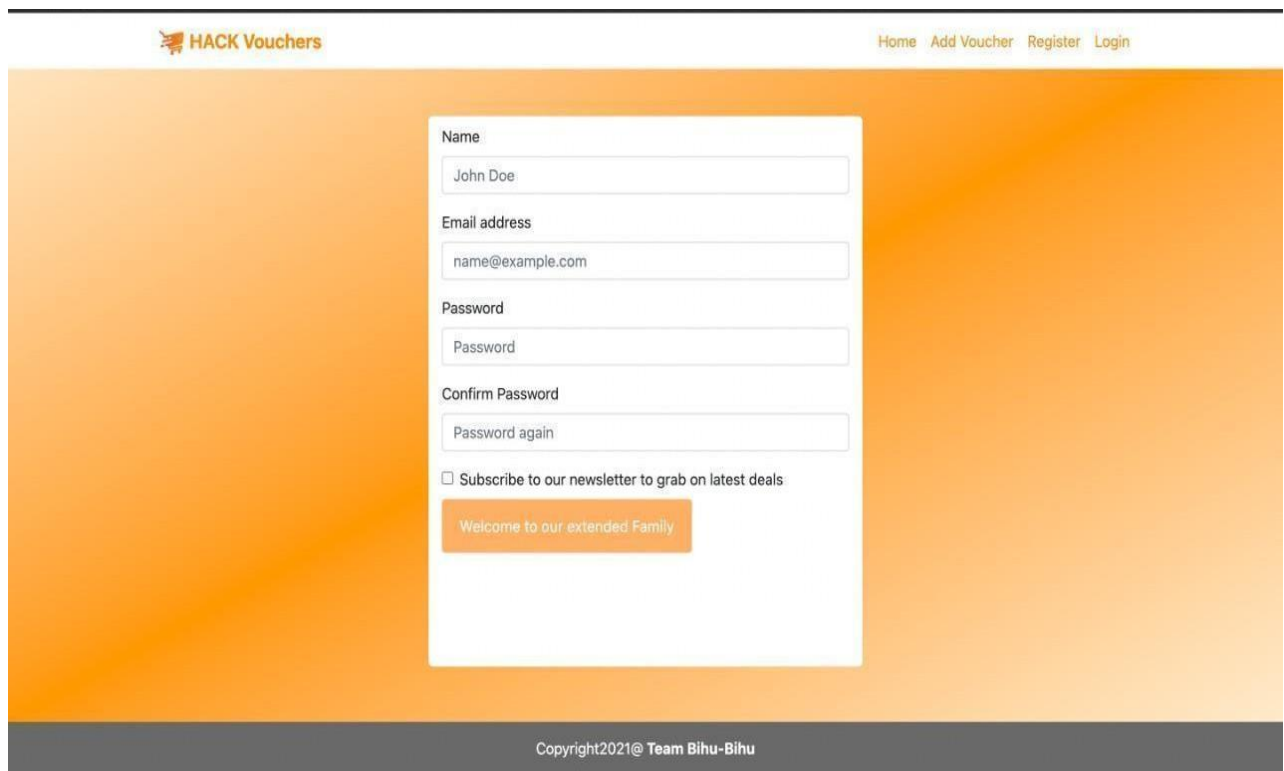
Fig. 6.1.6: Deployment Diagram



## CHAPTER 7 SYSTEM IMPLEMENTATION

### 7.1 Onboarding Flow

#### 7.1.1 Sign-up Flow



The screenshot displays the 'HACK Vouchers' sign-up interface. At the top left is the logo, and at the top right are navigation links: Home, Add Voucher, Register, and Login. The main content area has an orange gradient background. A white sign-up form is centered, containing the following fields: 'Name' (with 'John Doe' entered), 'Email address' (with 'name@example.com' entered), 'Password' (with 'Password' entered), and 'Confirm Password' (with 'Password again' entered). Below these fields is a checkbox labeled 'Subscribe to our newsletter to grab on latest deals'. At the bottom of the form is an orange button that says 'Welcome to our extended Family'. The footer of the page is dark gray and contains the text 'Copyright2021@ Team Bihu-Bihu'.

Figure 7.1 shows Signup screen UI

For user to use our website they have to registered on our website for that they have to fill a general detail by which we identify them and using which they login in our website. The detail that they have to fill are Name of User, Email address, Password and Confirm Password. Once user fill these details user are registered in our website.

## 7.1.2 Sign-in Flow

HACK Vouchers

Home Add Voucher Register Login

Email address

xyz@gmail.com

We'll never share your email with anyone else.

Password

\*\*\*\*\*

☐ Check me out

Glad to see you again, mate

Copyright2020@ Team Bihu-Bihu

Figure 7.2 shows sign in screen UI

Once a user is registered, they have to login/sign-in by using an email id and password that they provided at the time of registration in our website. If the detail that user provided is matched with the details that are in our database then it is allowed to login.

## 7.2 Dashboard

Dashboard is the page which will act like the main menu of the website. It has a feature showing the profile of the user, filter of vouchers, sign out, available coupon/vouchers, numbers of registered users, Number of vouchers, number of vouchers exchanged. Dashboard contains the links of all the pages like add voucher, profile.

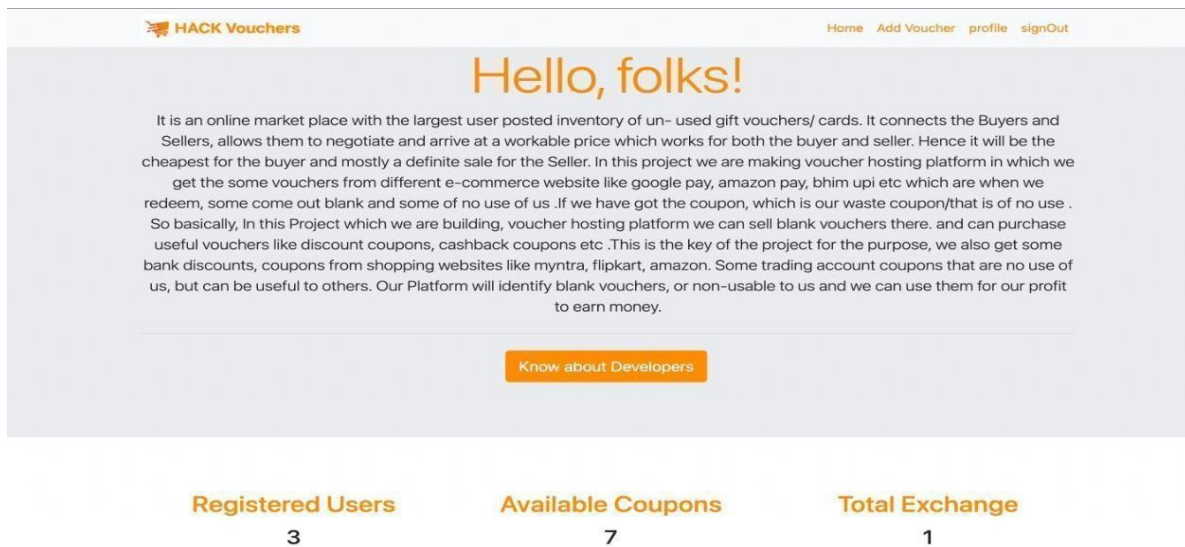


Figure 7.3 shows the Dashboard

## 7.2 Voucher

This section shows the details of vouchers like voucher title, voucher description, name of online store, price of voucher.

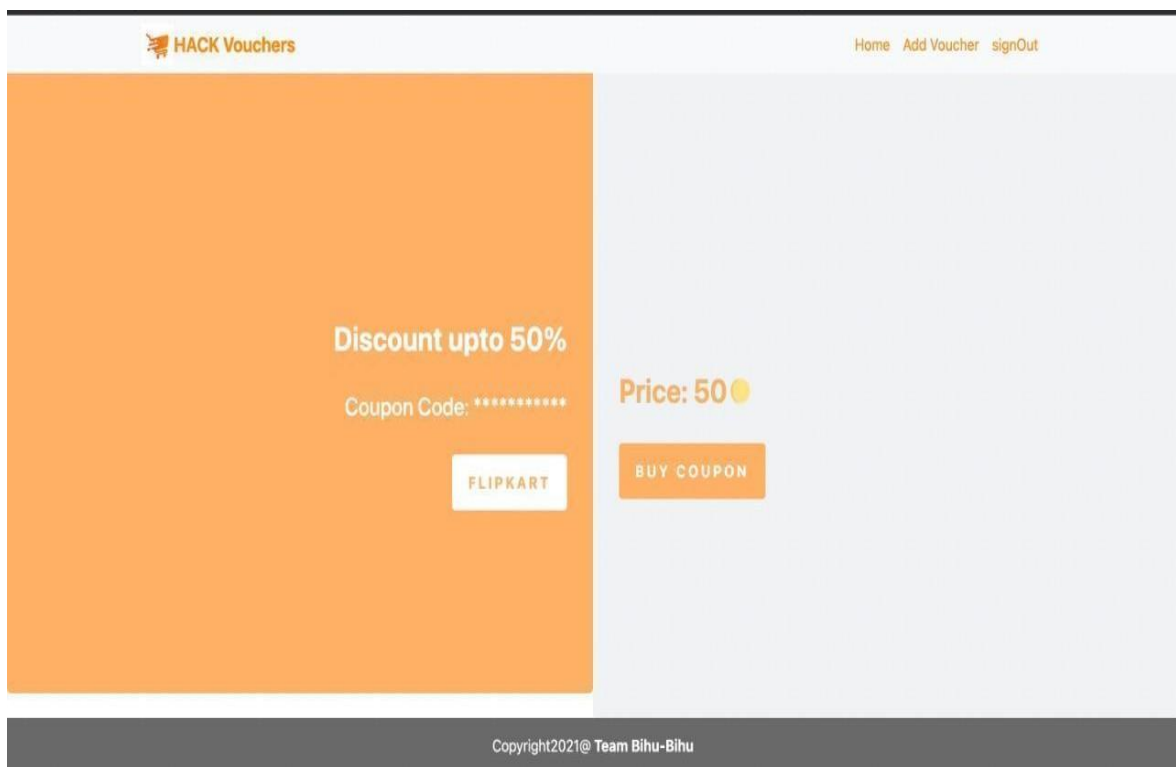


Figure 7.2 shows the voucher

## 7.2.1 Add Voucher

This section allows users to add their voucher which is to be sell .Users add the voucher by filling all the details in our “Seller Form” like Title ,name of Online store, Description, voucher code.

The screenshot displays the 'Add Voucher' form on the HACK Vouchers website. The form is a white box with a light orange border, centered on a light orange background. It contains the following fields and elements:

- Title:** A text input field containing '50% OFF...'
- Name of Online Store:** A text input field containing 'FLIPKART'
- Description:** A text input field that is currently empty.
- Mention any minimum cart limit (if applicable):** A text input field containing 'Enter in rupees'
- Voucher/Coupon code:** A text input field containing 'Enter code here'
- SUBMIT:** An orange button with white text.

The top navigation bar is white and contains the text 'Home profile signOut'. The footer is a dark grey bar with the text 'Copyright2020@ Team Bihu-Bihu'.

Figure 7.2.1 add the voucher

## 7.2.2 Buy Voucher

This section allows user to buy the vouchers that are available on the home page .For that they have to click on the card link and then the voucher is open with all detail and then on clicking on buy option it is added to user profile.

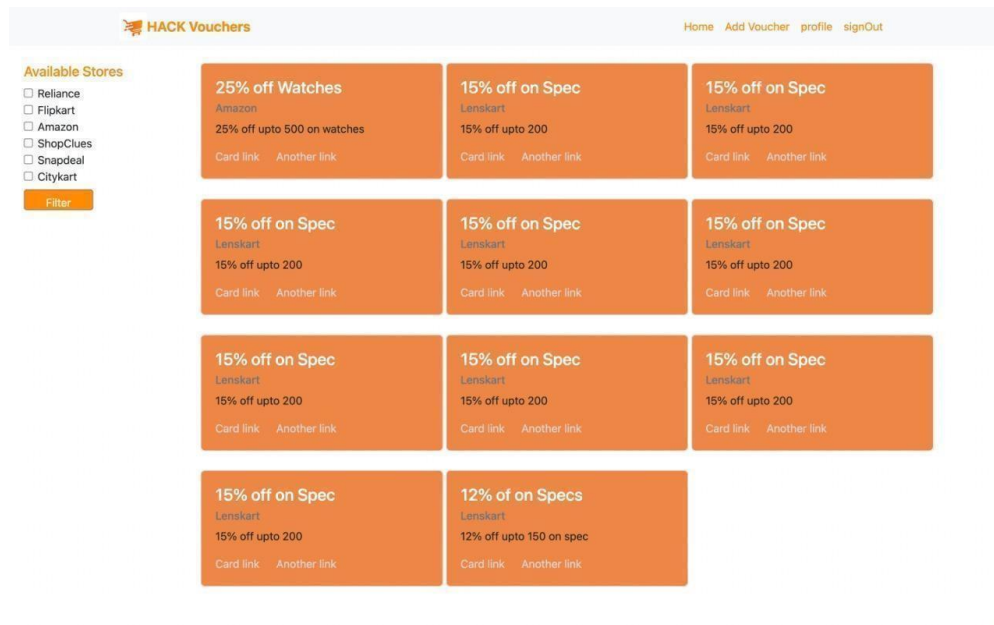


Figure 7.2.2 Buy voucher

## 7.3 Profile

This section shows the profile of users that contain user name, registered email, coupon hosted, coupon buy, coins in wallet.

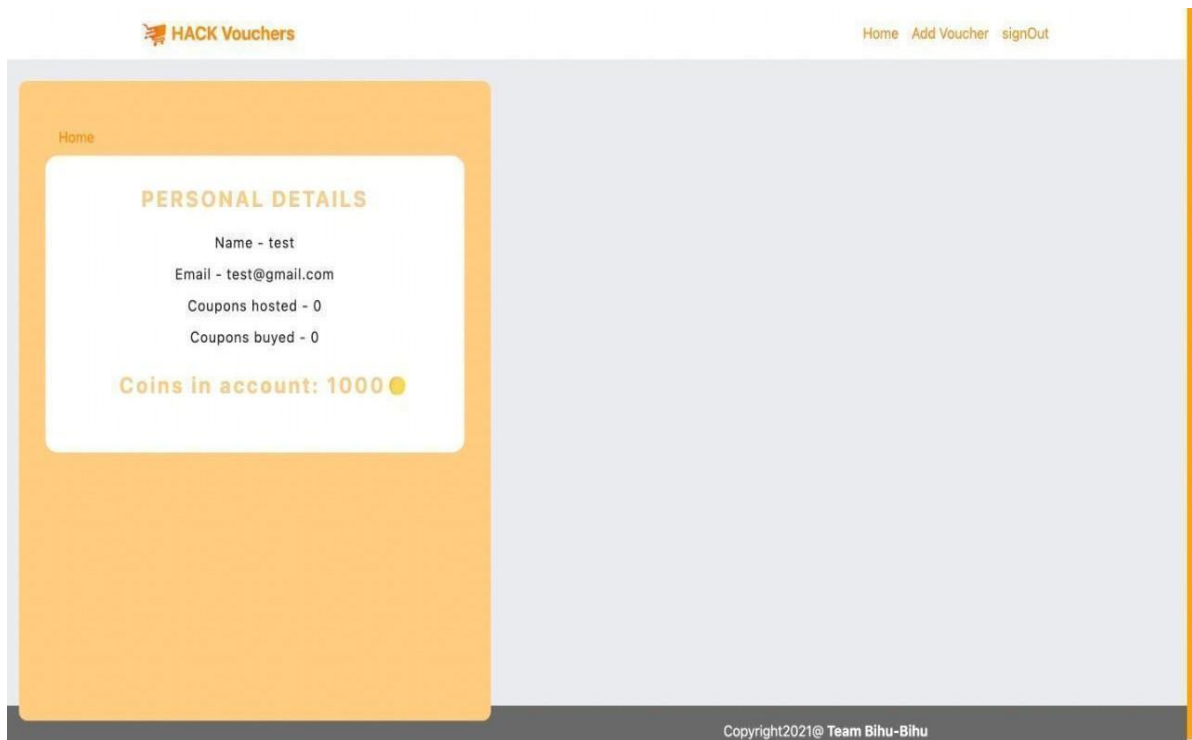


Figure 7.3.1 Buy voucher

## CHAPTER 8

### ER DIAGRAM

- An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram).
- An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes.
- In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.

### 8.1 Data Flow Diagram

The system design of our web application is described using data flow diagrams.

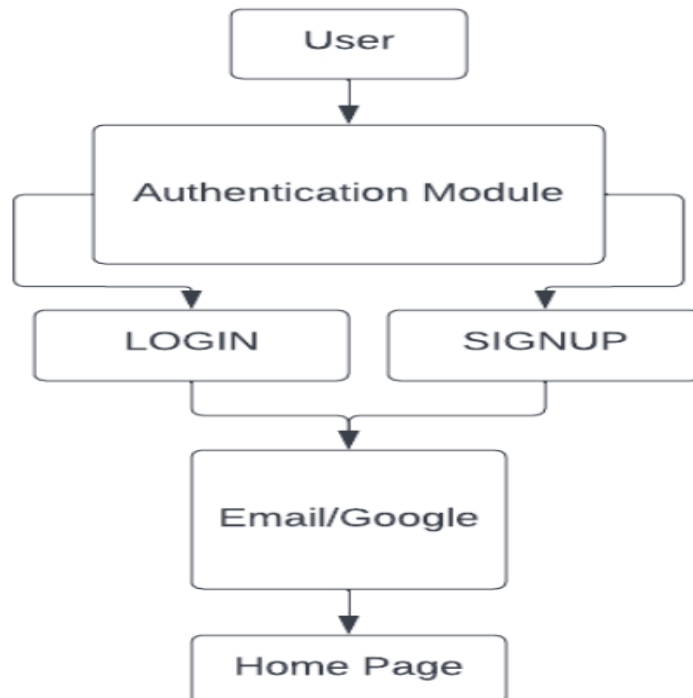


Fig 7.1- shows the data flow diagram of the user authentication module

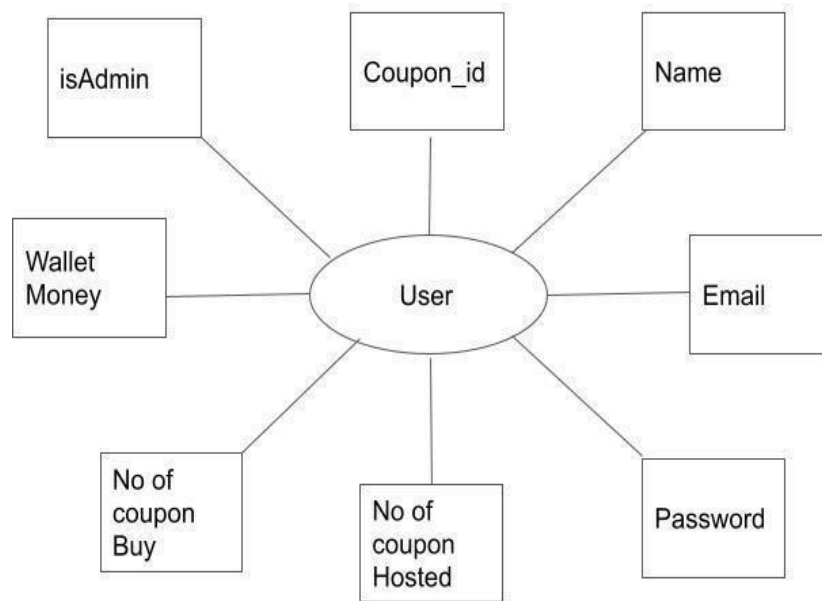


Fig 7.2 – User Schema diagram showing the various properties of use

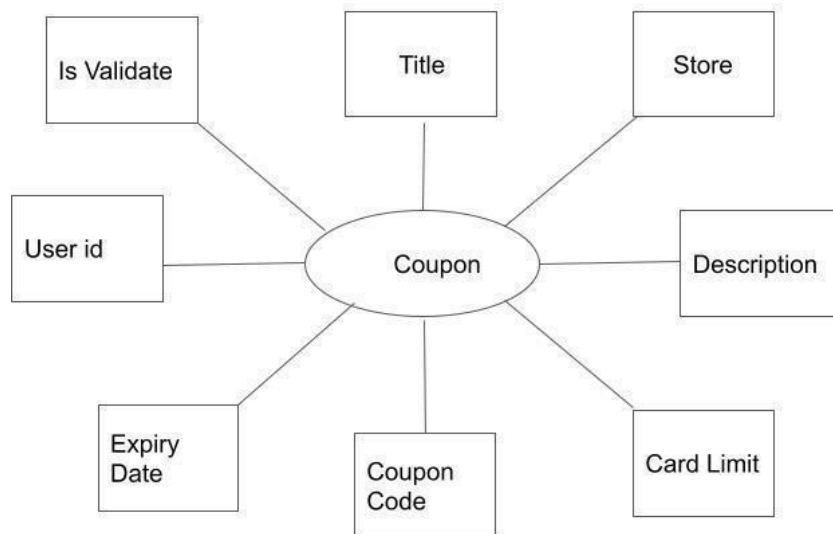


Fig 7.3 Coupon Schema showing the properties of coupon.

## CHAPTER 9

### SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

#### 9.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

HACK Vouchers

Home Add Voucher Register Login

Email address

test3@gmail.com

We'll never share your email with anyone else.

Password

\*\*\*\*

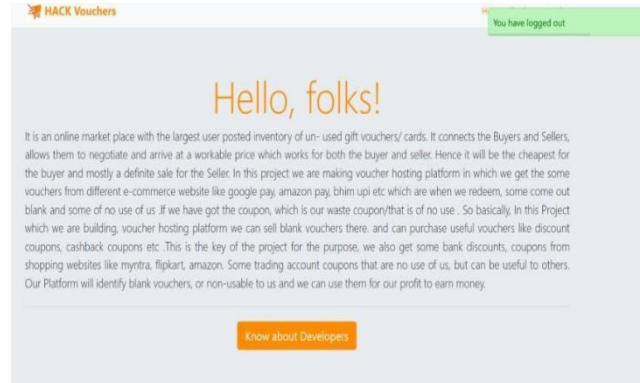
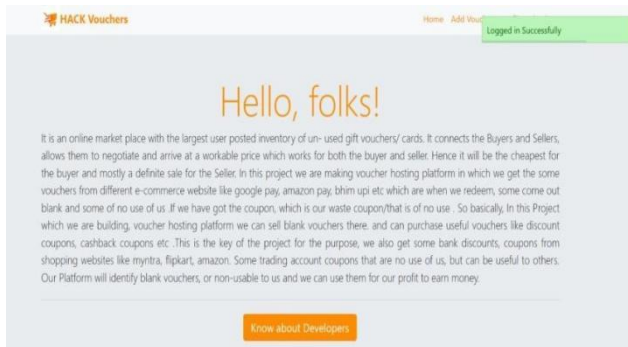
☒ Check me out

Glad to see you again, mate



## Features to be tested:

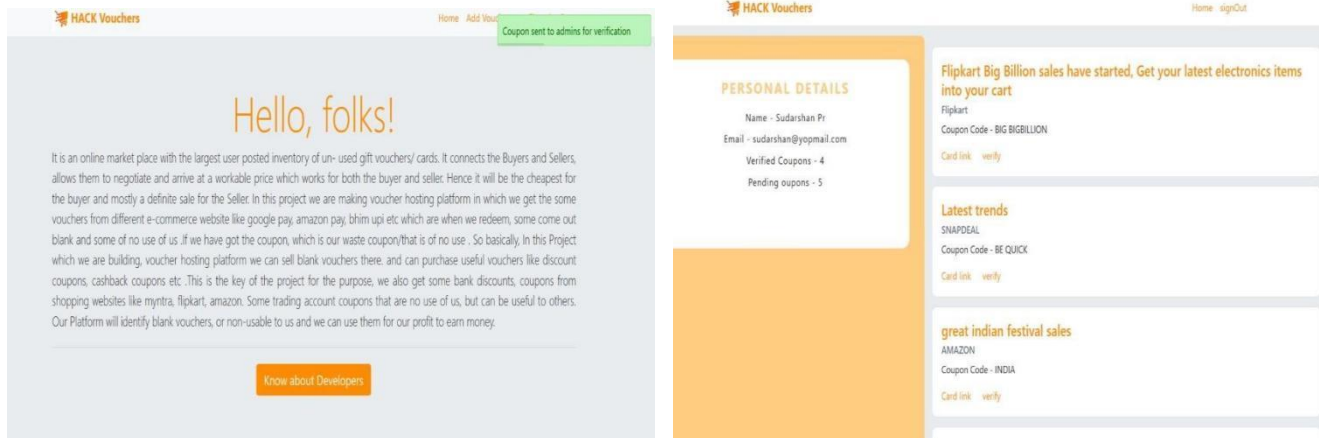
- Verify that the entries are of the correct format.
- No duplicate entries should be allowed.
- All links should take the user to the correct page.



## 9.2 INTEGRATION TESTING

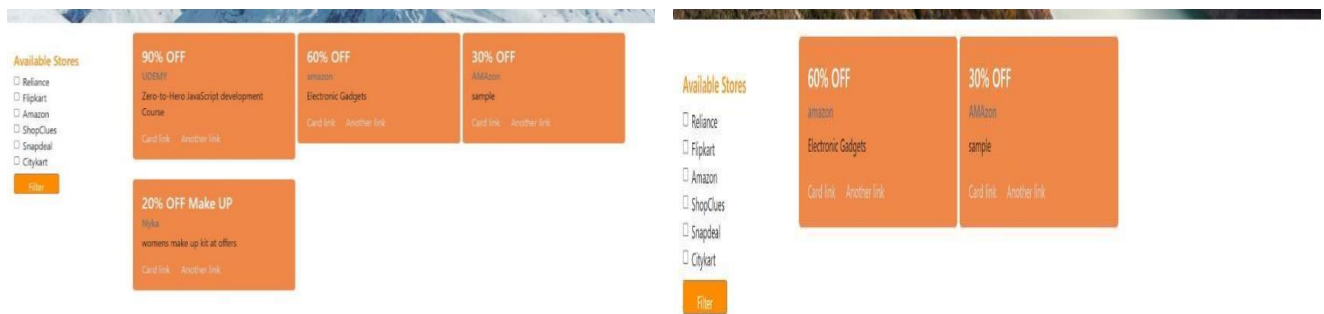
Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

A screenshot of the HACK Vouchers website showing a form to add a new voucher. The form is set against an orange background and contains the following fields: 'Title' (with '90% OFF Idols'), 'Name of Online Store' (with 'AMAZON'), 'Description' (with 'Festival sale'), 'Mention any minimum cart limit (if applicable):' (with '50'), and 'Voucher/Coupon code:' (with 'FEST123'). An orange 'SUBMIT' button is located at the bottom of the form.



## 9.3 SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.



### Features to be tested:

- System Testing evaluates the overall functionality and performance of a complete and fully integrated software solution.
- It tests if the system meets the specified requirements and if it is suitable for delivery to the end-users. This type of testing is performed after the integration testing and before the acceptance testing.
- The testers do not require more knowledge of programming to carry out this testing

### 9.4 ACCEPTANCE TESTING

- Acceptance Testing is a method of software testing where a system is tested for acceptability.
- The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not.
- This testing helps the project team to know the further requirements from the users directly as it involves the users for testing.

### CHAPTER 10 SAMPLE SOURCE CODE

#### Coupon.js source code

```
const express = require("express");
const routes = express.Router();
console.log("Routes has added");
const passport = require("../config/passport-local-strategy");
const User = require("../models/user");
const coupon = require("../models/coupon");
const { deleteOne } = require("../models/user");
const admins = ["nisar@gmail.com", "shivam@gmail.com", "pramod@gmail.com"];
routes.post("/create-coupon", async (req, res) => {
  var obj = await new coupon({
    title: req.body.title,
    store: req.body.store,
    description: req.body.description,
    cardLimit: req.body.cardLimit,
    couponCode: req.body.couponCode,
    user: res.locals.user._id,
    isFiltered: true,
  });
  obj.save();
  res.locals.user.coupen_hosted = res.locals.user.coupen_hosted + 1;
  res.locals.user.save();
  req.flash("success", "Coupon sent to admins for verification");
  return res.redirect("/home");
});
routes.get("/main/:id", async (req, res) => {
  let COUPON = await coupon.findById(req.params.id);
  res.render("bootmain", {
    COUPON: COUPON,
    link: "www." + COUPON.store + ".com",
  });
});
```

```
});  
routes.get("/buy-coupon/:id", async (req, res) => {  
  let COUPON = await coupon.findById(req.params.id);  
  res.locals.user.COUPONS.push(COUPON);  
  res.locals.user.coins = res.locals.user.coins - COUPON.cardLimit;  
  res.locals.user.coupen_buied = res.locals.user.coupen_buied + 1;  
  COUPON.isPurchased = true;  
  let user = await User.findById(COUPON.user);  
  user.coins = user.coins + COUPON.cardLimit;  
  user.save();  
  COUPON.save();  
  res.locals.user.save();  
  return res.redirect("/home");  
});  
routes.post("/filter", async (req, res) => {  
  let data = await coupon.find({ isPurchased: false, isFiltered: true });  
  var totalPendingCoupons = 0;  
  for (let i = 0; i < coupon.length; i++) {  
    if (data[i].isVerified) {  
      totalPendingCoupons = totalPendingCoupons + 1;  
    }  
  }  
  console.log(req.body.companyName);  
  let userSize = await (await User.find()).length;  
  var filter;  
  var x = [];  
  var y = req.body.companyName;  
  if (!Array.isArray(y)) {  
    x.push(y);  
    filter = x;  
  } else {  
    filter = req.body.companyName;  
  }  
  console.log(filter);
```

```
res.render("boothome", {
  isAuthenticated: req.isAuthenticated(),
  COUPON: data,
  userSize: userSize,
  couponCount: await (await coupon.find({ isPurchased: false })).length,
  coupoExchange: await (await coupon.find()).length,
  filteredCompany: filter,
  filtred: true,
  user: res.locals.user,
  totalPendingCoupons: totalPendingCoupons,
  totalVerifiedCoupons: coupon.length - totalPendingCoupons,
  totalAdmins: admins.length,
  isSearched: false,
});
});
routes.get("/verify/:id", async (req, res) => {
  let COUPON = await coupon.findById(req.params.id);
  COUPON.isVerified = true;
  COUPON.save();
  // Todo: check if the user is admin or not to restrict the access to call the function.
  res.redirect("back");
});
module.exports = routes;
```

### User\_Data\_format.js

```
const mongoose = require('mongoose') ;
const UserSchema = new mongoose.Schema({
  name :{
    type : String ,
    required : true ,
    unique : true
  },
  email : {
```

```
    type : String,
    required : true
  },
  password : {
    type : String ,
    required : true
  },
  coupon_hosted : {
    type : Number,
    default : 0
  },
  coupon_buied : {
    type : Number,
    default : 0
  },
  coins : {
    type : Number,
    default : 1000
  },
  COUPONS : [
    {
      type : mongoose.Schema.Types.ObjectId,
      ref : 'coupon'
    }
  ],
  is_profile_pic_uploaded : {
    type : Boolean,
    default : false
  },
  isAdmin : {
    type : Boolean,
    default : false
  },
  userInterest : [
```

```
    {
      type : String
    }
  ]
});
const User = mongoose.model('User' , UserSchema) ;
module.exports = User ;
```

### Coupon\_data\_format.js

```
const mongoose = require('mongoose') ;
const User = require('../models/user')
const couponSchema = new mongoose.Schema({
  title : {
    type : String,
    required : true
  },
  store :{
    type : String,
    required : true
  },
  description : {
    type : String,
    required : true
  },
  cardLimit : {
    type : Number,
    required : true
  },
  couponCode : {
    type : String,
    required : true
  },
  user : {
```



```
    type : mongoose.Schema.Types.ObjectId,
    ref : 'user'
  },
  isPurchased : {
    type : Boolean ,
    default : false
  },
  isFiltered : {
    type : Boolean,
    required : true
  },
  isVerified : {
    type : Boolean,
    default : false
  }
})
const coupon = mongoose.model('coupon' , couponSchema) ;
module.exports = coupon ;
```

### Mongoose.js

```
const mongoose = require('mongoose') ;
mongoose.connect('mongodb+srv://nisaar:Nidsar@cluster0.ztmue.mongodb.net/Hackvoucher_collection?retryWrites=true&w=majority') ;
const db = mongoose.connection ;
db.on('error' , console.error.bind(console , "error connecting to db")) ;
db.once('open' , function(){
  console.log('successfully connected to database') ;
}) ;
```

### Middleware.js

```
module.exports.setFlash = (req , res , next) => {  
  res.locals.flash = {  
    'success' : req.flash('success') ,  
    'error' : req.flash('error')  
  }  
  next() ;  
}
```

### Passport\_local\_strategy.js

```
const passport = require('passport') ;  
const LocalStrategy = require('passport-local').Strategy;  
const User = require('../models/user') ;  
passport.use(new LocalStrategy({  
  usernameField : 'email',  
  passReqToCallback : true  
},  
(req , email , password , done) => {  
  User.findOne({ email : email } , (err , user) => {  
    if(err) {  
      req.flash('error' , err)  
      return done(err) ;  
    }  
    if(!user || user.password !== password) {  
      req.flash('error' , 'Invalid Username/Password') ;  
      return done(null , false) ;  
    }  
    return done(null , user) ;  
  })  
})  
);
```

```
passport.serializeUser((user , done) => {
  done(null , user.id) ;
});
passport.deserializeUser((id , done) => {
  User.findById(id , (err , user) =>{
    if(err){
      console.log('error in finding user --> passport') ;
      return done(err) ;
    }
    return done(null , user) ;
  })
});
```

```
passport.checkAuthentication = (req , res , next) => {
  //console.log(req);
  if(req.isAuthenticated()) {
    return next() ;
  }
  console.log('###') ;
  return res.redirect('/sign-in') ;
}
```

```
passport.setAuthenticatedUser = (req , res , next) =>{
  if(req.isAuthenticated()){
    // console.log(req.user)
    res.locals.user = req.user ;
  }
  next() ;
}
module.exports = passport ;
```

BootMain.js

```
{% load static %}
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="description" content="">
  <meta name="author" content="">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>HackVoucher</title>
  <link rel="shortcut icon" href="" type="image/x-icon">
  <link rel="apple-touch-icon" href="">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <!-- Link to your css file -->
  <link rel="stylesheet" href="">
  <style>
    ::-webkit-scrollbar {
      width: 12px;
    }

    /* Track */
    ::-webkit-scrollbar-track {
      background: #f1f1f1;
    }

    /* Handle */
    ::-webkit-scrollbar-thumb {
      background: orange;
    }

    /* Handle on hover */
```

```
::-webkit-scrollbar-thumb:hover {
    background: darkorange;
}
</style>
</head>

<body>
    <!-- Start coding here -->
    <nav class="navbar navbar-expand-md navbar-light bg-light fixed-top" style="background-color:
white;">
        <div class="container">
            
            <a class="navbar-brand" href="/home" style="color: darkorange"><strong>HACK
Vouchers</strong></a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarResponsive">
                <ul class="navbar-nav ml-auto">
                    <li class="nav-item active">
                        <a class="nav-link" href="/home" style="color: darkorange">Home<span class="sr-
only">(current)</span></a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/add-voucher" style="color: darkorange">Add Voucher</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/sign-out" style="color: darkorange">signOut</a>
                    </li>
                </ul>
            </div>
        </div>
    </div>
```

```
</nav>

<div class="jumbotron" style="width: 50%; height: 656px; margin-bottom: 0; display: inline-block;
float: left; background-color: darkorange; opacity: 0.7; color: white;">

  <h2 style="text-align: right; margin-top:
35%;"><strong><%=COUPON.description%></strong></h2>

  <br>

  <% if(COUPON.isPurchased) { %>

    <h4 style="text-align: right;">Coupon Code:<%= COUPON.couponCode
%><strong></strong></h4>

  <% }else { %>

    <h4 style="text-align: right;">Coupon Code: *****<strong></strong></h4>

  <% }%>

  <br>

  <a href="https://<%= link %>"><button type="button" class="btn btn-primary" style="float:
right; background-color: white; color: darkorange; outline:none; border: none; padding: 15px 20px;
letter-spacing: 3px; text-transform:
uppercase;"><strong><%=COUPON.store%></strong></button></a>

</div>

<div class="jumbotron" style="width: 50%; height: 656px; margin-bottom: 0; display: inline-block;
color: darkorange; opacity: 0.7;">

  <h2 style="text-align: left; margin-top: 39%; display: inline-block;"><strong>Price:
<%=COUPON.cardLimit%></strong></h2>

  <br>

  <br>

  <a href="/buy-coupon/<%= COUPON._id %>"><button type="button" class="btn btn-primary"
style="background-color: darkorange; color: white; outline:none; border: none; padding: 15px 20px;
letter-spacing: 0.2em;"><strong>BUY COUPON</strong></button></a>

</div>

<!-- Coding End -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
```

## HACK VOUCHER

---

```
crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
</body>
</html>
```

## CHAPTER - 11

### RESULTS AND SCREEN LAYOUTS

These are some of the results that are observed in our project

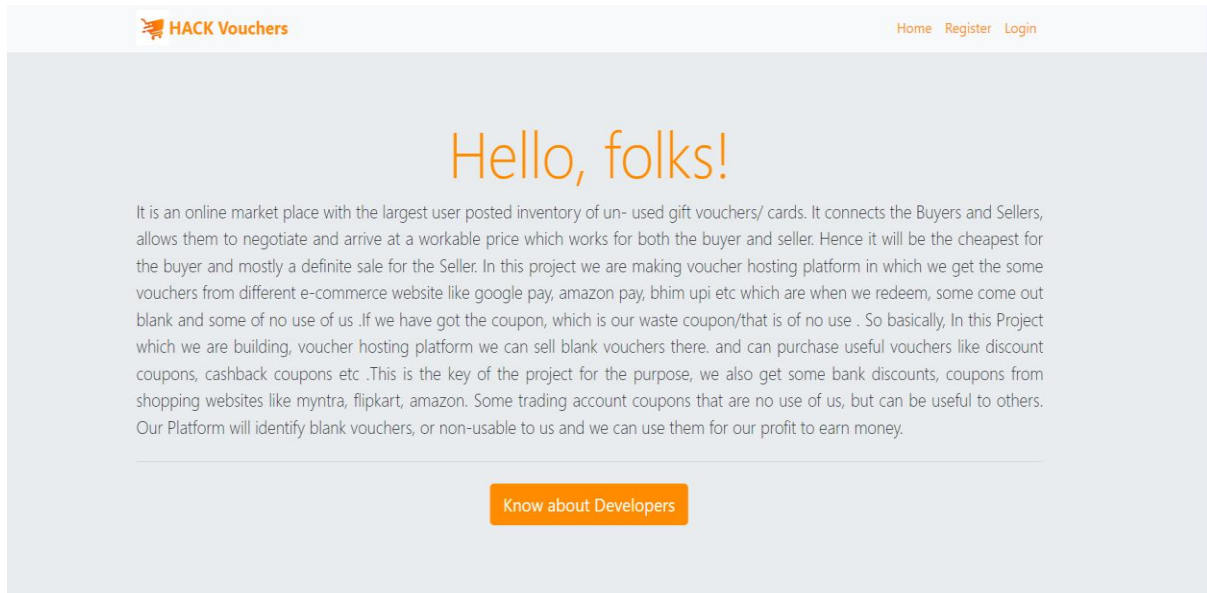


Fig 11.1 - Home Page of HackVoucher

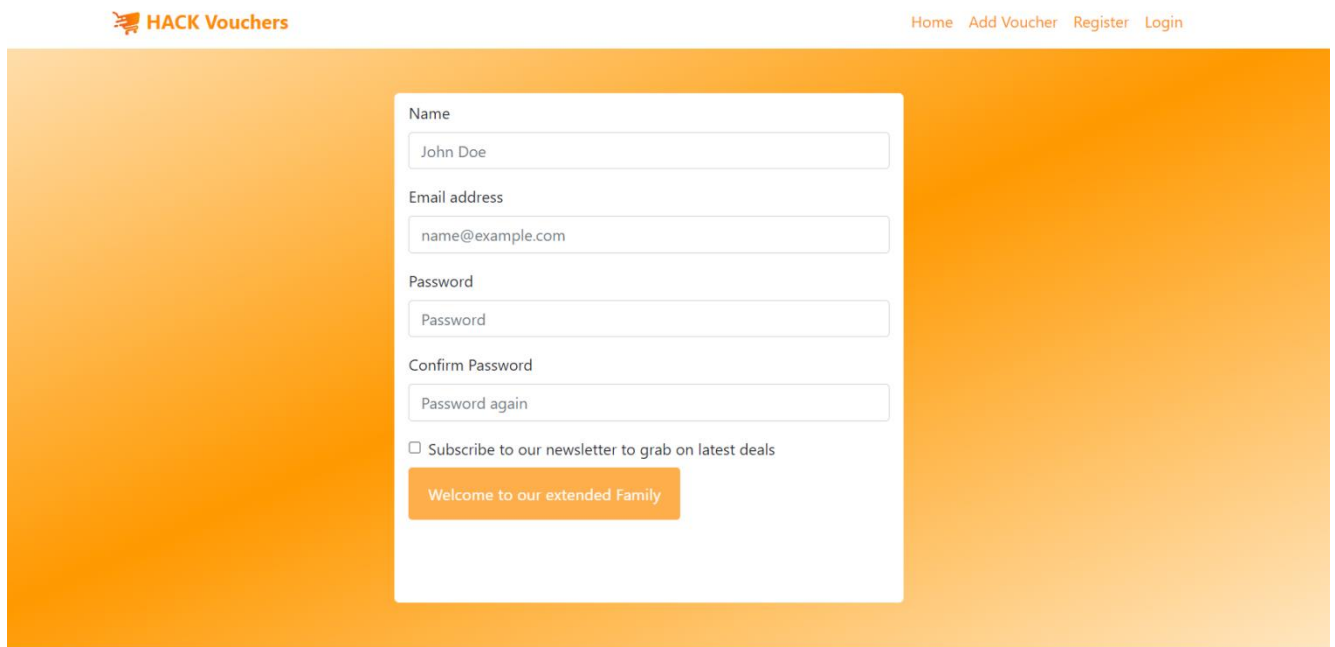


Fig 11.2 - Register page for new User



The screenshot shows the login page for the HACK Vouchers application. The header includes the HACK Vouchers logo and navigation links: Home, Add Voucher, Register, and Login. The main content area has an orange gradient background. A white login form is centered, containing fields for Email address and Password. Below the password field is a checkbox for 'Check me out' and an orange button labeled 'Glad to see you again, mate'.

HACK Vouchers

Home Add Voucher Register Login

Email address

Enter email

We'll never share your email with anyone else.

Password

Password

☐ Check me out

Glad to see you again, mate

Fig 11.3 - Login Page for Existing Users

The screenshot shows the 'Add Voucher' page in the HACK Vouchers application. The header includes the HACK Vouchers logo and navigation links: Home, profile, and signOut. The main content area has an orange gradient background. A white form is centered, with fields for Title, Name of Online Store, Description, and Voucher/Coupon code. There is also a field for 'Mention any minimum cart limit (if applicable):' with a placeholder 'Enter in rupees'. An orange 'SUBMIT' button is at the bottom of the form.

HACK Vouchers

Home profile signOut

Title

50% OFF....

Name of Online Store

FLIPKART

Description

Mention any minimum cart limit (if applicable):

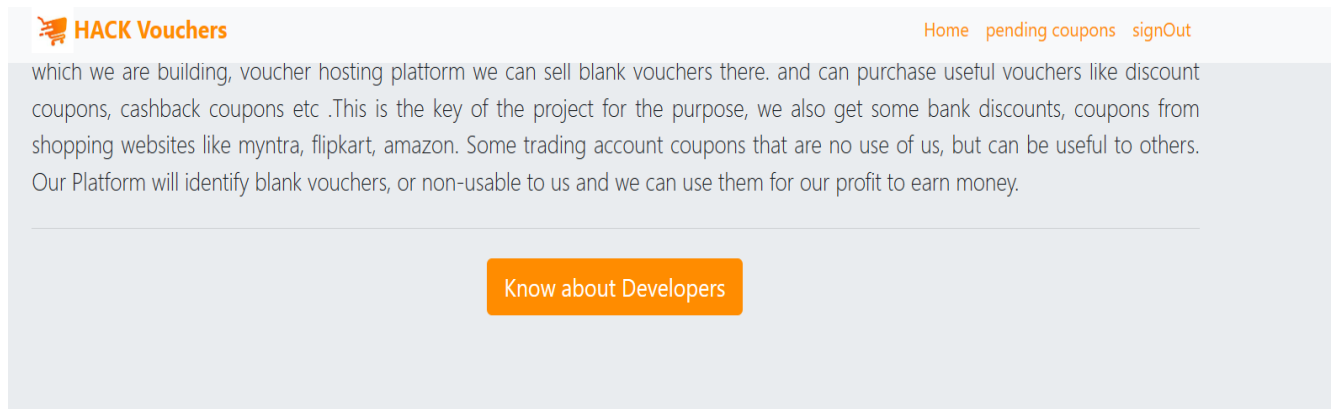
Enter in rupees

Voucher/Coupon code:

Enter code here

SUBMIT

Fig 11.4 - Add Voucher page to add New Vouchers into the platform



**Verified Coupons**

9

**Pending Coupons**

4

**Total Admins**

3

Fig 11.5 - Application showing Real-Time data to all users

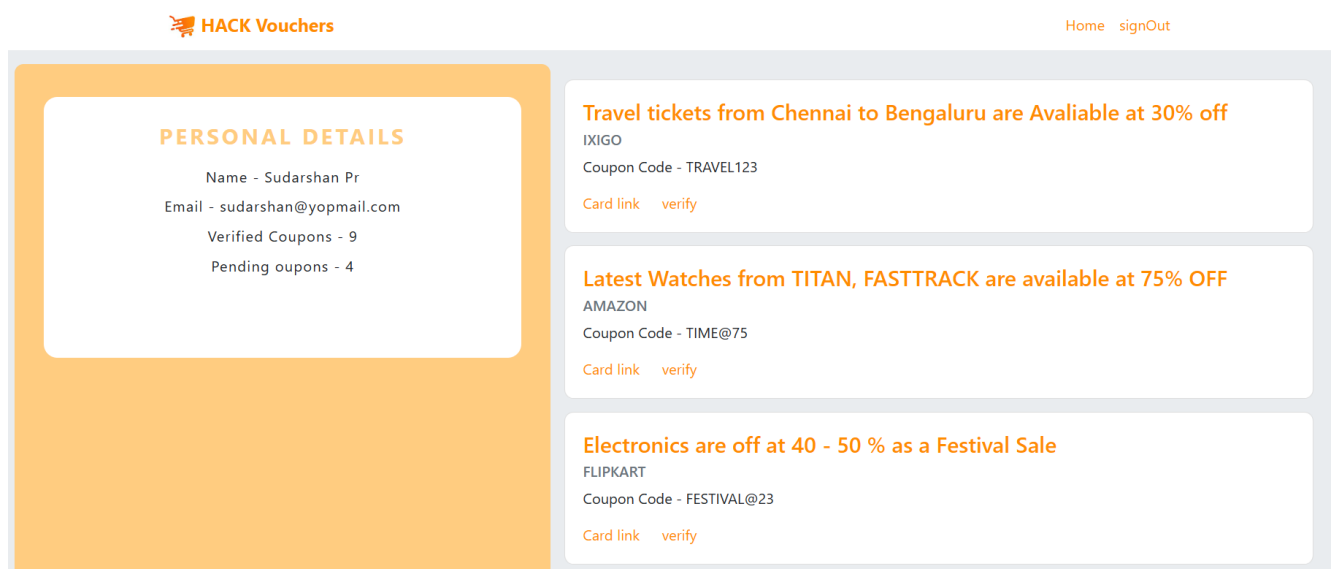



Fig 11.6 - Admin Profile Showing Pending coupons to get verified

 HACK Vouchers

[Home](#) [pending coupons](#) [signOut](#)

Available Stores

- ☐ Reliance
- ☐ Flipkart
- ☐ Amazon
- ☐ ShopClues
- ☐ Snapdeal
- ☐ Citykart

Filter

60% OFF

amazon

Electronic Gadgets

Card link Another link

30% OFF

AMAZon

sample

Card link Another link

70% Electronics

Flipkart

Flipkart Big Billion sales have started, Get your latest electronics items into your cart

Card link Another link

45% DISCOUNT

SNAPDEAL

Latest trends

Card link Another link

20% OFF Make UP

Nyka

womens make up kit at offers

Card link Another link

40%

AMAZON

great indian festival sales

Card link Another link

90% OFF Idols

Amazon

Festival sale

Card link Another link

40% Gadgets

SNAPDEAL

Factory Stock

Card link Another link

40% off

SVCE

college coupon

Card link Another link

Fig 11.7 - Showing all the available coupons in store to buy

Available Stores

- ☐ Reliance
- ☐ Flipkart
- ☐ Amazon
- ☐ ShopClues
- ☐ Snapdeal
- ☐ Citykart

Filter

60% OFF

amazon

Electronic Gadgets

Card link Another link

30% OFF

AMAZon

sample

Card link Another link

40%

AMAZON

great indian festival sales

Card link Another link

90% OFF Idols

Amazon

Festival sale

Card link Another link

Fig 11.8 - Shows all the filtered coupons belonging to AMAZON store

S V COLLEGE OF ENGINEERING, DEPARTMENT OF CSE

58

### **CHAPTER 12**

### **CONCLUSION**

The end result of the mentioned methodology and planned action will be a highly interactive UI with working functionalities of a Hack -voucher, which will keep user data secure and will be able to handle customized coupon/gift card search for a user. This web application aims to eliminate the wastage of unused vouchers/gift cards. We also get some bank discounts, coupons from shopping websites like myntra, flipkart, amazon. Some trading account coupons that are of no use to us, but can be useful to others. Our Platform provide users coins in exchange of money that is our profit to earn money.

### CHAPTER 13 FUTURE ENCHANCEMENTS

Hack -Voucher is very useful for the users because in pandemic time all the e-commerce companies gives gift cards and vouchers for purchasing their products and also we getcashbacks by payments apps like google pay, phone pay , paytm etc. Now they found the baseof customers so now they give gifts cards and vouchers that are oftenly unusable for us .So here we provide a web application platform where users can sell their unused gift card and vouchers and purchase usable vouchers and gift cards.

We can also include a section for negotiation where users can negotiate on purchasing and selling of vouchers and gift cards.

- This feature is available to only logged in users.
- Choose the gift card you would to Negotiate the price and click on “Negotiate”
- It will open a window, on the Left side of the Text Area, there is a “Negotiate” button, click that to open the Negotiate box
- Enter the amount that you want to negotiate.

We would also develop android apps and launch on play store. android app makes more interactive.

UI and easy to use for users

We can also implement filter such a way that user view voucher of their interest and also get a notification on their profile and email when someone added a voucher of their interest.

As we have less time, we could not implement all the features but in future we implement all the above-mentioned points plus add other features that are given as feedback by faculties.

## CHAPTER 14 BIBLIOGRAPHY

### 14.1 REFERENCES

- <https://www.persistencemarketresearch.com/market-research/gift-card-market.asp>
- INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH  
VOLUME 8, ISSUE 08, AUGUST 2019 ISSN 2277-8616
- Volume: 57 issue: 4, page(s): 739-754  
Article first published online: April 27, 2020; Issue published: August 1, 2020.
- Public opinion on Security of Data from companies, *November 2019*, Brooke Auxier, Lee Rainie, Pew Research Center. Online Learning
- Wikipedia(<https://en.wikipedia.org/wiki/>)
- tutorials on point. (2020). SDLC-agile model-tutorialspoint. Retrieved from tutorials point.com

### 14.2 WEBSITES

<https://www.clevergirlfinance.com/blog/the-best-coupon-websites/>  
<https://medium.com/swlh/learn-how-to-build-a-full-stack-application-using-node-js-and-react-in-2021-part-1-121fa6762a69>  
<https://www.freecodecamp.org/news/develop-deploy-first-fullstack-web-app/>  
<https://medium.com/>  
<https://codingheroes.io/>  
<https://learn.mongodb.com/>