PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Software Engineering Project

Automated Attendance System

TEAM 04-J section

Veeresh R.G -PES1UG20CS609

Swathi Rupali N.V-PES1UG20CS600

Shristi Nadakatti-PES1UG20CS592

Phaneesh S-PES1UG20CS584

Identifying software life cycle model (SDLC), Agile methodology useused

Software Life Cycle Model: Iterative

We followed the Iterative model approach to implement/ develop our product.

Initial implementation starts from a skeleton of the product by simulating a prototype of our product .We came up with a basic idea on how we wanted our website to look like which was reiterated multiple times to achieve the best one. Using the Iterative SDLC model helped us identify requirements and visualize the solution. This also provided support for risk mitigation, incremental investment, reduced rework and feature creeps.

Despite the above advantages, each phase was rigid with overlaps in implementation.

Agile methodology: SCRUM

Scrum is a framework of rules, roles, events, and artifacts used to implement Agile projects. We developed our product using an iterative approach, consisting of sprints that typically lasted one to two weeks. This approach ensured that our team delivered a version of the product regularly.

SRS:

Functional requirements:

- Detection and identification of the face
- Mark attendance for the corresponding recognised faces.
- Being able to view the attendance in a web app.
- Export attendance via API..

Interface requirements:

- Will include a good UI interface through which users will be able to interact with the software

Non-functional requirements:

- Data integrity
- The software must be reliable
- Scalable and portable

Process model:

- Software processes is way for designing, implementing and testing a software product
  i.e. All the way from requirement analysis of the product to the deployment of the software.

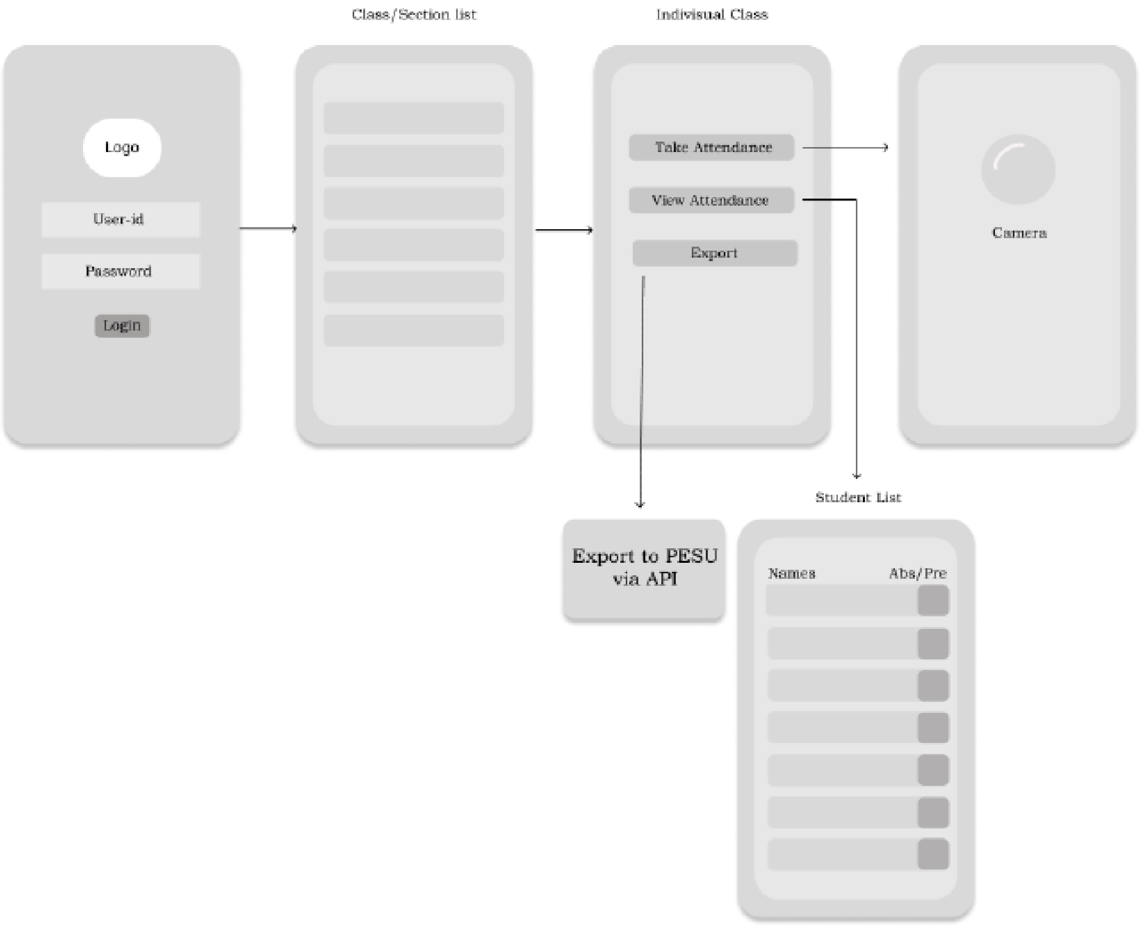
- Software process models are abstract representations of this entire development process

- There are various models that can be followed. These models have various stages and represent a sequential flow of events / actions.

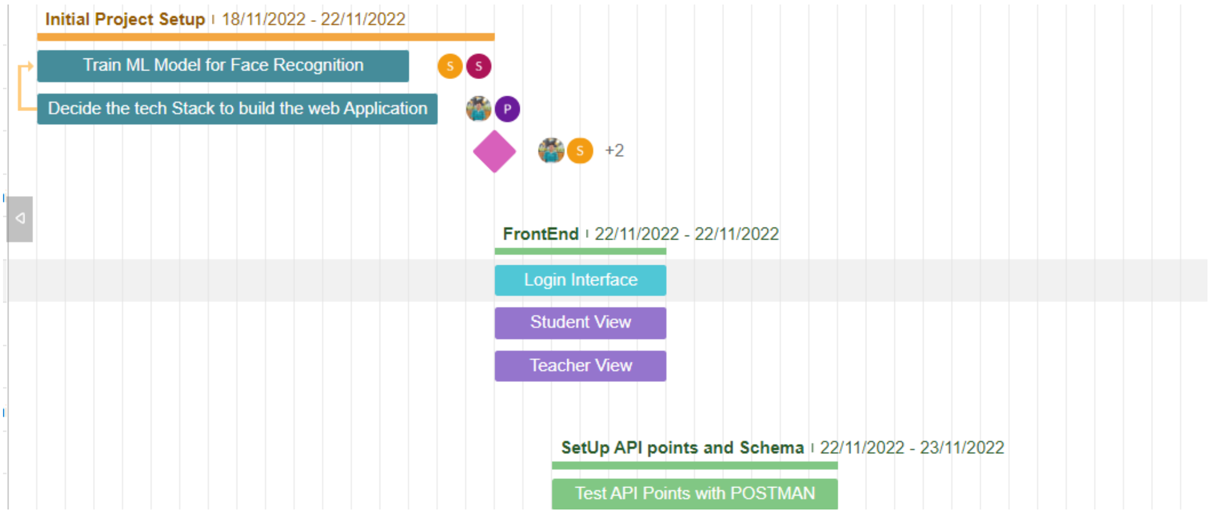
- Choosing the right model for a project is very crucial to achieve the end-product and objectives as effectively as possible.

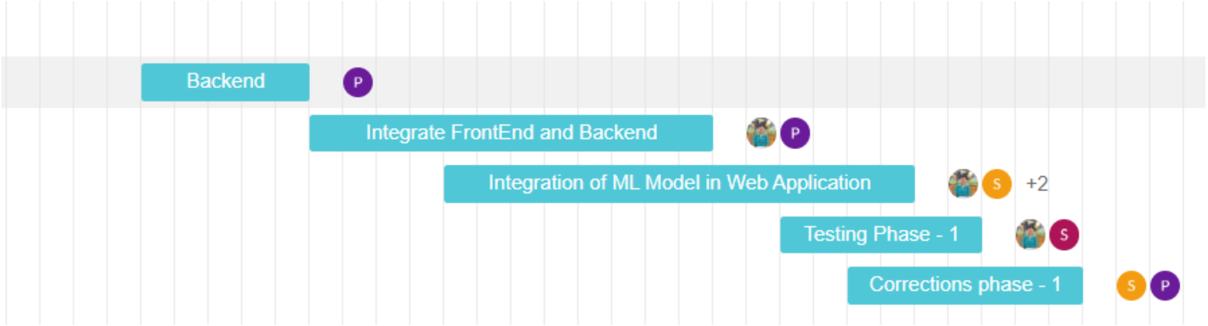
- There are various ways of choosing process models:
    - Project requirements
    - Project size
    - Cost
    - Complexity of the project
    - Number of developers on board
- For our project, we are choosing the scrum methodology which implements the agile model.
- So, every feature of the project is developed in a single or multiple iterations called sprints. In our case each sprint session will last for atleast1 week.
- Why scrum?
    - Helps us deliver features / changes faster
    - Being agile, accepts and tries implementing user’s and stakeholders reviews.
    - Since we are a small team, managing using scrum would be very efficient and less-challenging.
    - A large project is divided into features / modules that a team can achieve (implement) for each sprint session

Prototype Built:

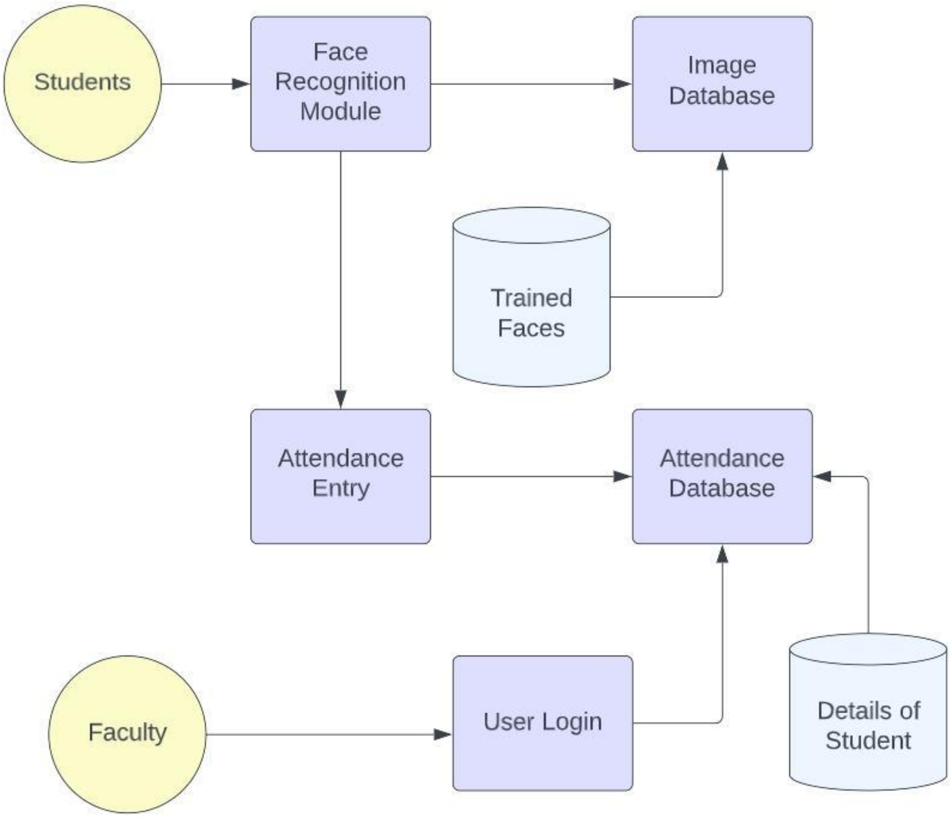


Gantt Chart:


Initial Project Setup | 18/11/2022 - 22/11/2022
Train ML Model for Face Recognition
S S
Decide the tech Stack to build the web Application
P
S +2
FrontEnd | 22/11/2022 - 22/11/2022
Login Interface
Student View
Teacher View
SetUp API points and Schema | 22/11/2022 - 23/11/2022
Test API Points with POSTMAN


Backend
P
Integrate FrontEnd and Backend
P
Integration of ML Model in Web Application
S +2
Testing Phase - 1
S
Corrections phase - 1
S P

Architecture:



Students:
The students face is captured after which the attendance is marked for that particular student.

Face Recognition Module:
Here the student's face undergoes training in the model where different significant points are detected after which the face is mapped or classified as that particular student's name.

Image Database:
Here images of students are present.

Trained Database:
In this database we have all the trained images which allows us to map attendance.

Attendance Entry:
In this particular phase , once the student's face is recognized it is mapped to the json file where the attendance is marked for that particular student.

Attendance Database:
Here we keep a track of the attendance of different classes in a particular university/school.

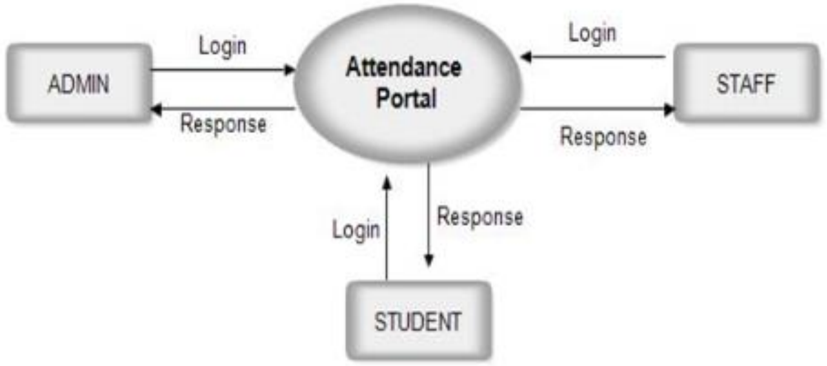
Faculty:
The faculty is given access to the web application where he/she can capture an image of the class for attendance.

User Login:
The faculty uses their login credentials to log in to their particular account where they have access to all the classes they teach. Here the class is selected and a picture is captured. Once the picture is captured the attendance is automatically mapped from each face in the picture to the database containing their personal details.

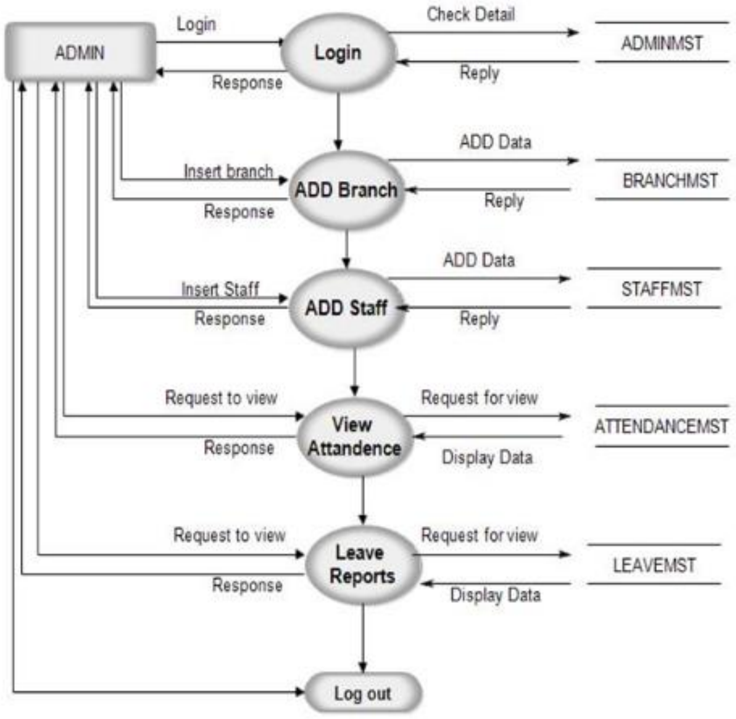
Details of Student:
This contains details of every student in the class along with the attendance. Once the attendance is mapped it is loaded onto the attendance database which allows the student to access their attendance history.

Data Flow Diagram

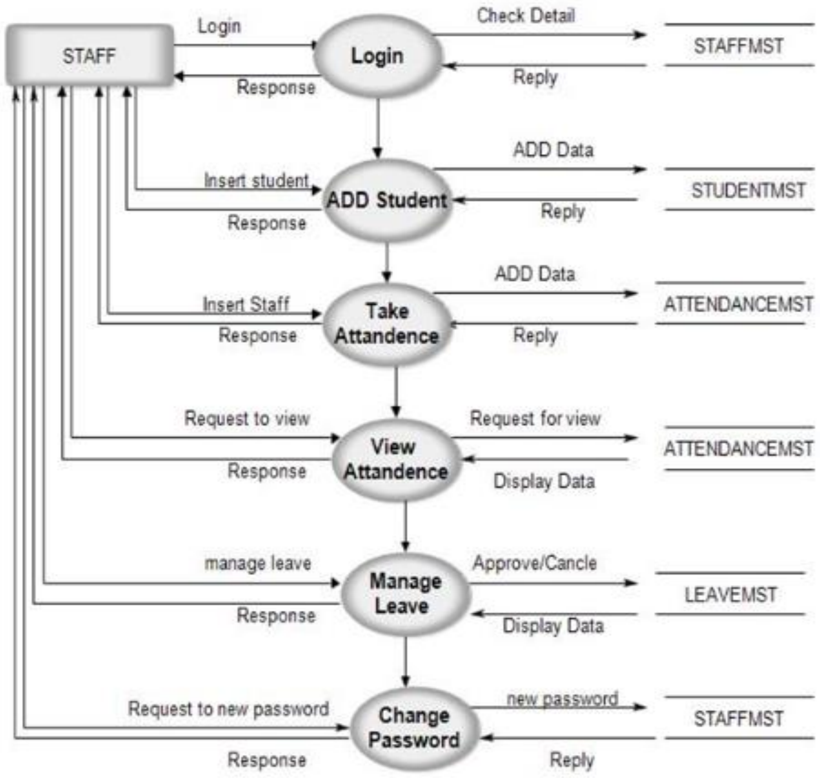
0 - Level DFD : Context Level



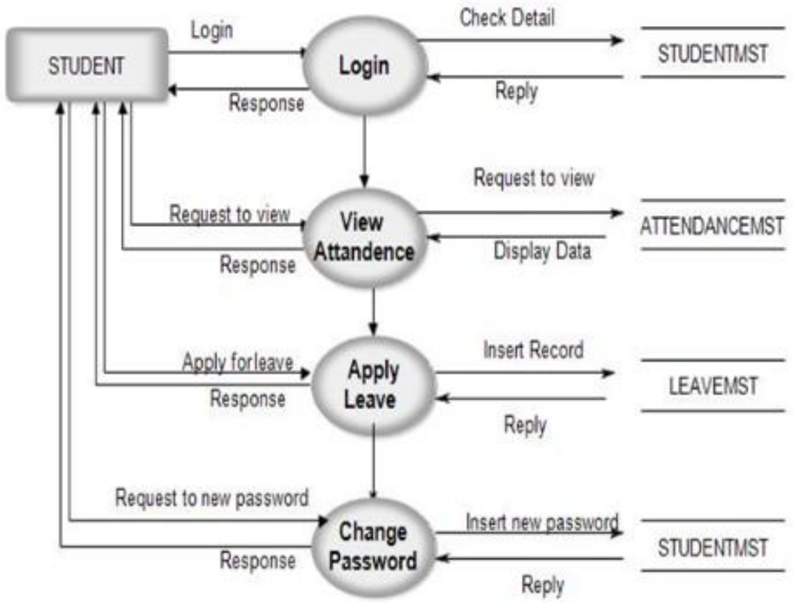
ADMIN - Data Flow Diagram



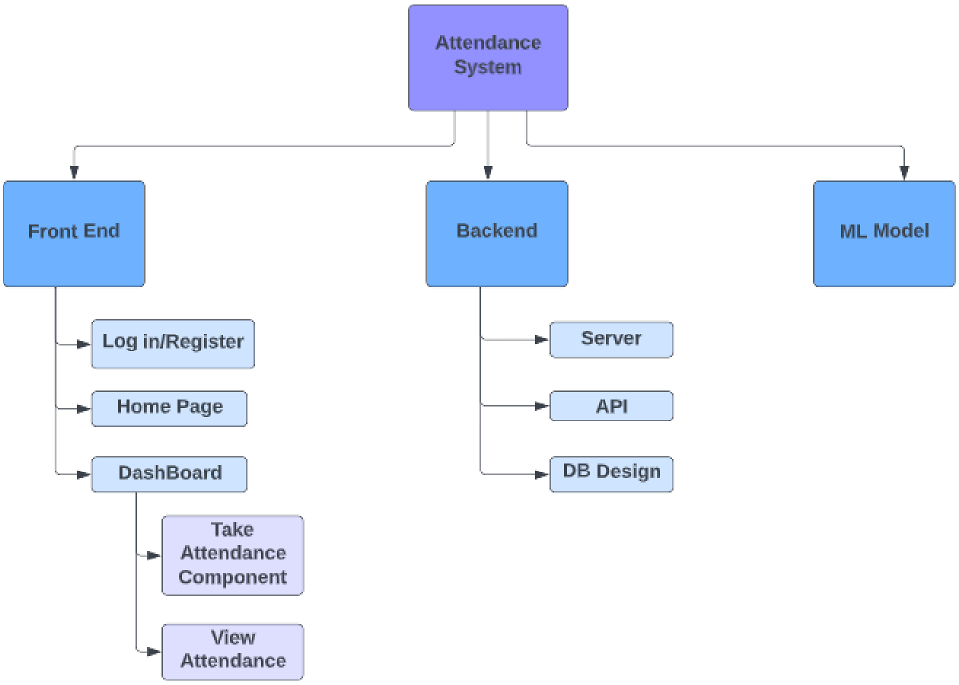
STAFF - Data Flow Diagram



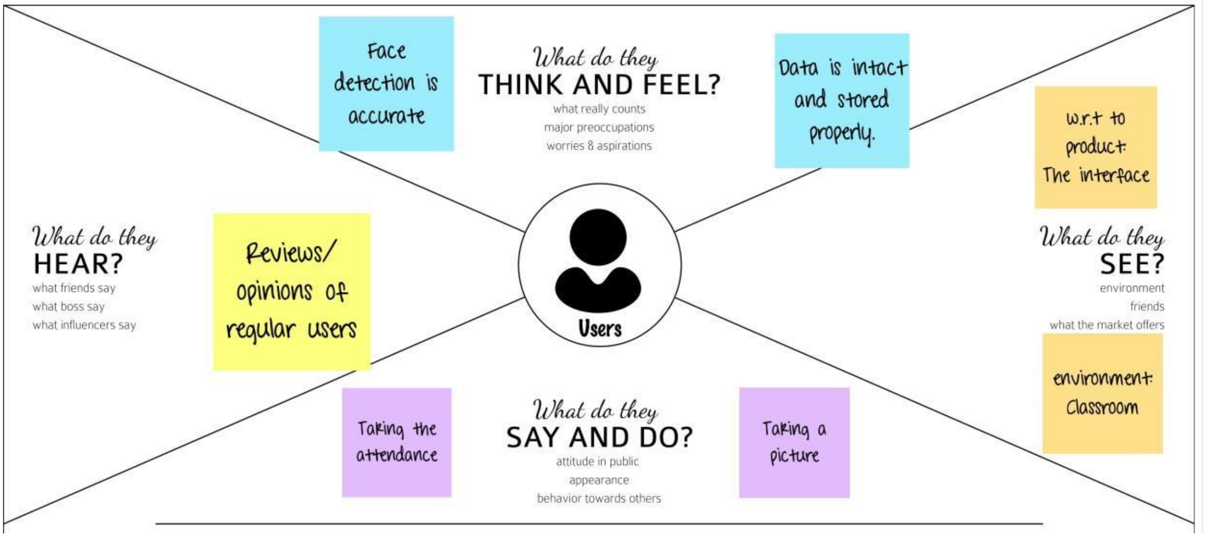
STUDENT - Data Flow Diagram



Design:



Empathy Map:



Explain Coding practices/standards used

- Focus on code readability and maintainability.
    - Write as few lines as possible
    - Use appropriate naming conventionshttps://github.com/phaneesh707/SE_project
    - Segments blocks
- Reduce complexity us much as possible
- Divide the code into components to maximize reusability
- Documentation is maintained and codes are well commented
- Don’t use a single identifier for multiple purposes.
- Turn daily backup’s into an instinct
- Formalize Exception handling
    - Use try - catch

SCM environment used like GitHub and any SCM concepts such as Branch Management and Versioning used for the project.

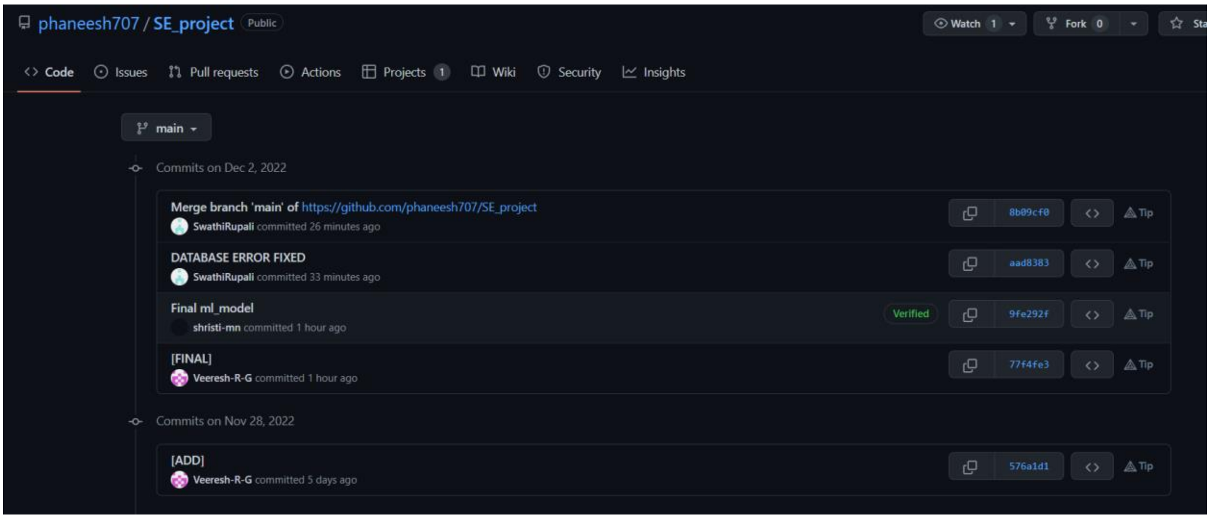
Software Configuration Management (SCM) is a software engineering discipline consisting of standard processes and techniques often used by organizations to manage the changes introduced into its software products. SCM helps in identifying individual elements and configurations,tracking changes, and version selection, control, and baselining.

SCM is a process to systematically organize, manage and control changes in documents, code and other entities that constitute a software product.

Github- We decided to use Github for managing, organizing our project for the following reasons:

- Git is free and open source software for distributed version control.
    - Tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development.
- Access control
- Bug tracking
- Software feature requests
- Task management

GitHub Repository:



Commits on Dec 2, 2022

Merge branch 'main' of https://github.com/phaneesh707/SE_project
SwathiRupali committed 26 minutes ago
8b09cf0

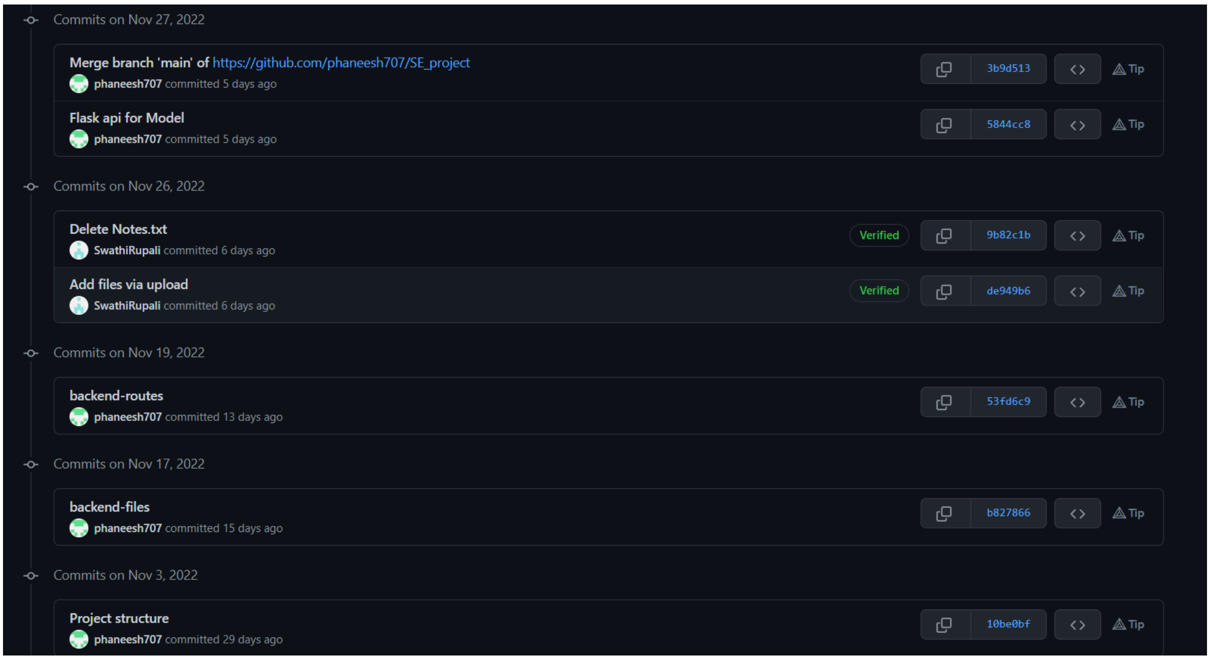
DATABASE ERROR FIXED
SwathiRupali committed 33 minutes ago
aad8383

Final ml_model
shristi-mn committed 1 hour ago
Verified
9fe292f

[FINAL]
Veeresh-R-G committed 1 hour ago
77f4fe3

Commits on Nov 28, 2022

[ADD]
Veeresh-R-G committed 5 days ago
576a1d1



Commits on Nov 27, 2022

Merge branch 'main' of https://github.com/phaneesh707/SE_project
phaneesh707 committed 5 days ago
3b9d513

Flask api for Model
phaneesh707 committed 5 days ago
5844cc8

Commits on Nov 26, 2022

Delete Notes.txt
SwathiRupali committed 6 days ago
Verified
9b82c1b

Add files via upload
SwathiRupali committed 6 days ago
Verified
de949b6

Commits on Nov 19, 2022

backend-routes
phaneesh707 committed 13 days ago
53fd6c9

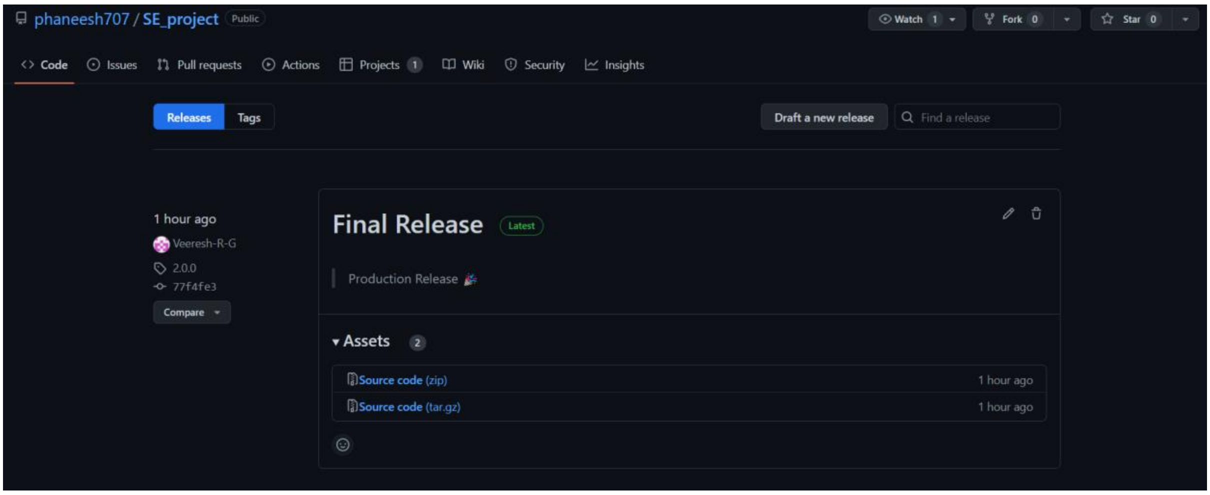
Commits on Nov 17, 2022

backend-files
phaneesh707 committed 15 days ago
b827866

Commits on Nov 3, 2022

Project structure
phaneesh707 committed 29 days ago
10be0bf

Releases & Versions:



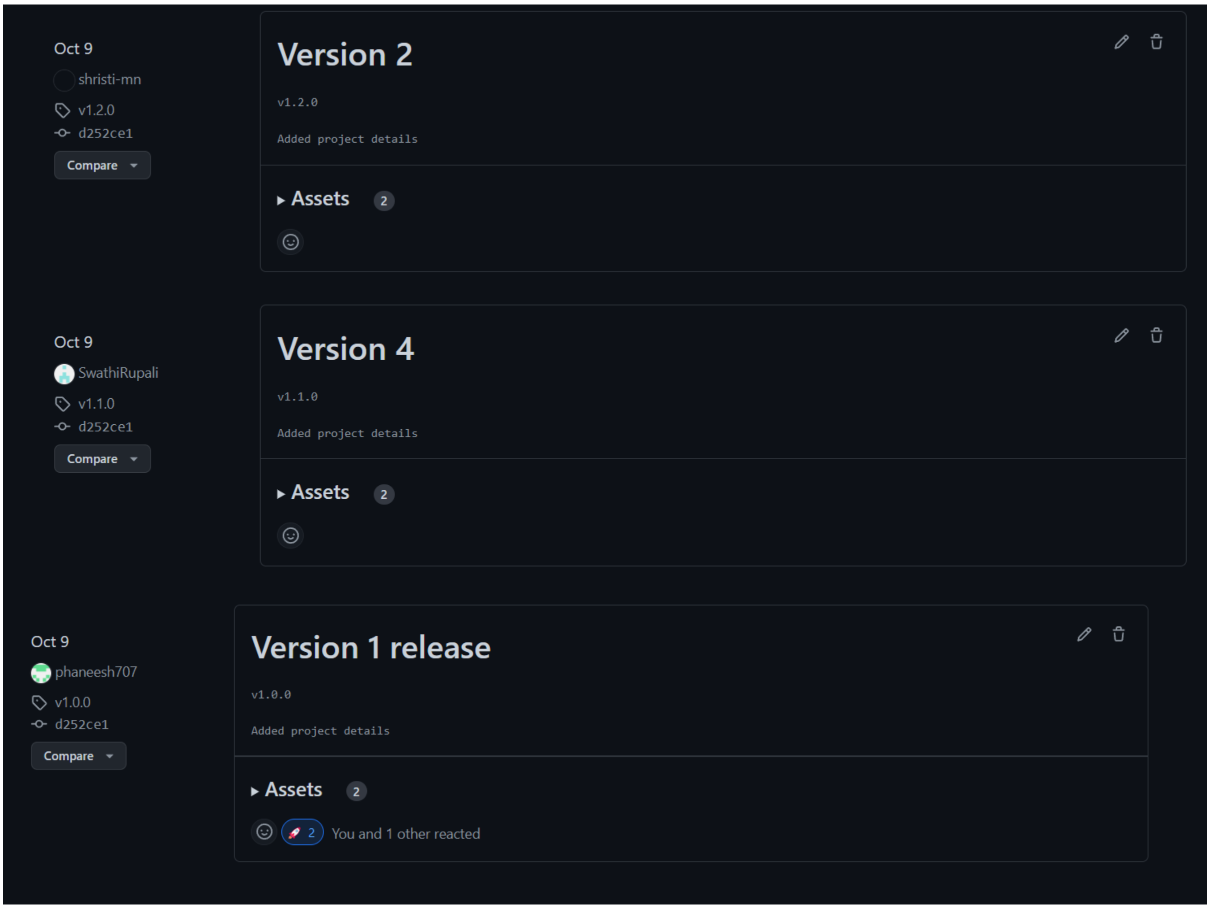
1 hour ago
Veeresh-R-G
2.0.0
77f4fe3

Final Release Latest

Production Release 🎉

Assets 2

Source code (zip)	1 hour ago
Source code (tar.gz)	1 hour ago



Oct 9
shristi-mn
v1.2.0
d252ce1

Version 2

v1.2.0

Added project details

Assets 2

Oct 9
SwathiRupali
v1.1.0
d252ce1

Version 4

v1.1.0

Added project details

Assets 2

Oct 9
phaneesh707
v1.0.0
d252ce1

Version 1 release

v1.0.0

Added project details

Assets 2

🚀 2 You and 1 other reacted

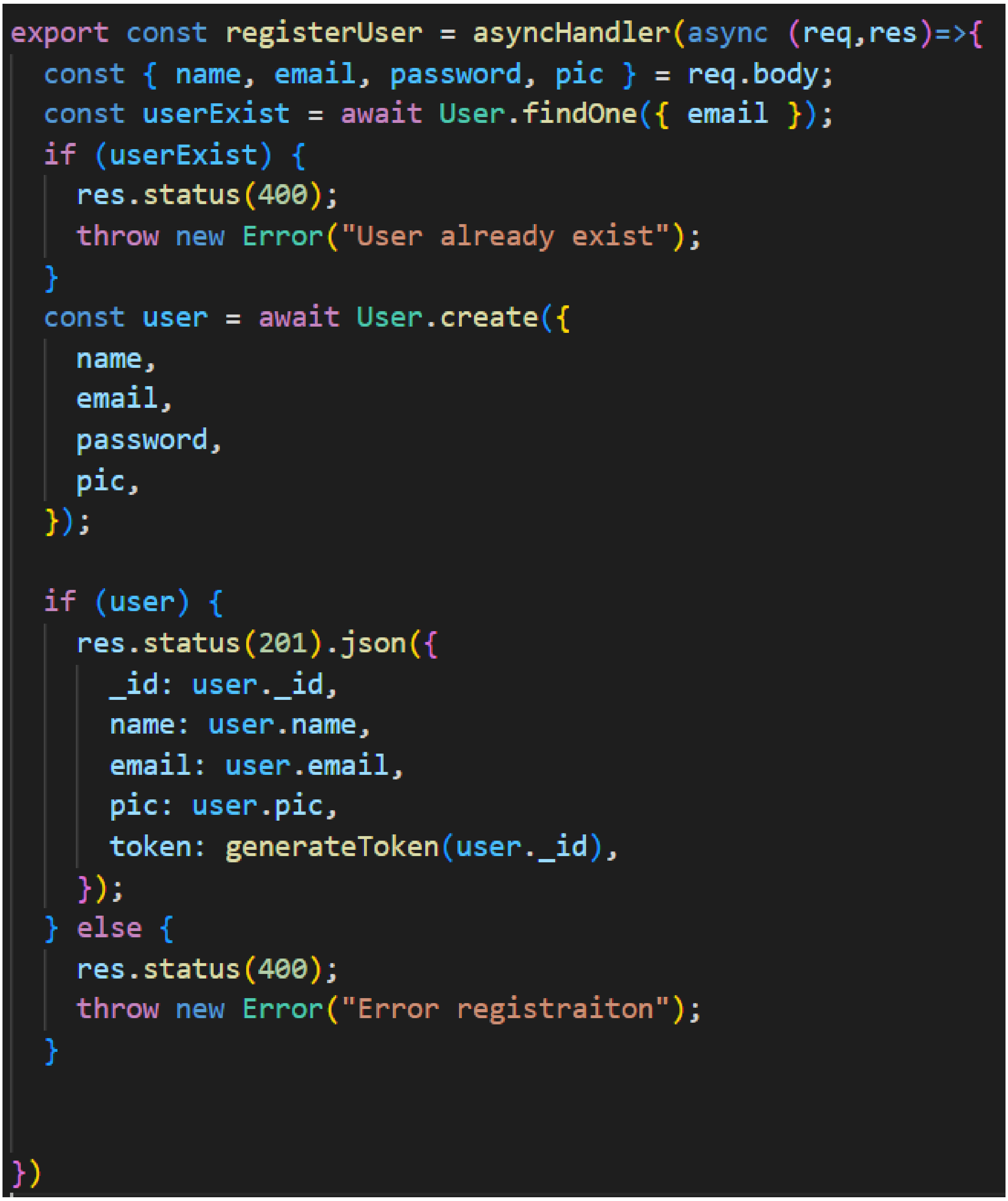
[Github Link]

Test strategy, test plan, Test Suite, Test Cases

Cyclomatic complexity is a measurement to determine the stability and level of confidence in a program. It measures the number of linearly-independent paths through a program module

For static testing, we have used a unit of code from the backend which handles the logic for the user's registration.

Code and its corresponding flowchart for calculating cyclomatic complexity

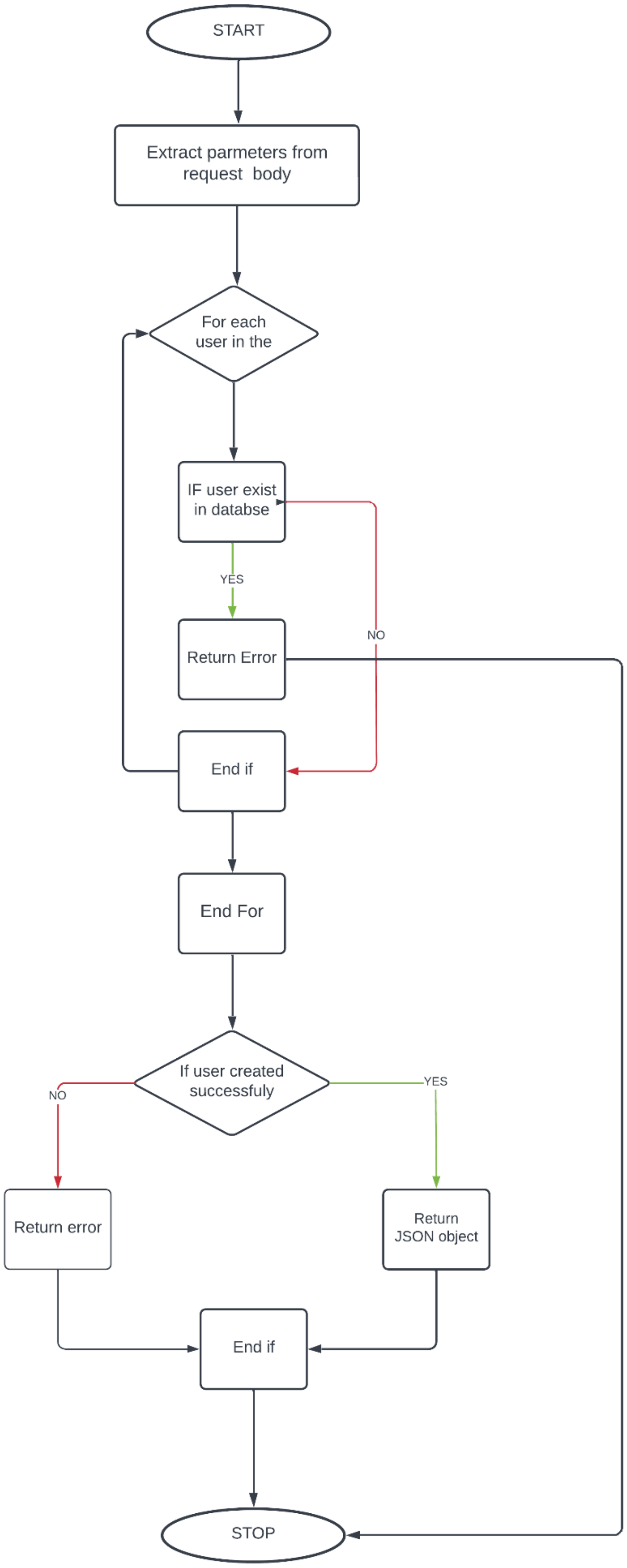


```
export const registerUser = asyncHandler(async (req,res)=>{
  const { name, email, password, pic } = req.body;
  const userExist = await User.findOne({ email });
  if (userExist) {
    res.status(400);
    throw new Error("User already exist");
  }
  const user = await User.create({
    name,
    email,
    password,
    pic,
  });

  if (user) {
    res.status(201).json({
      _id: user._id,
      name: user.name,
      email: user.email,
      pic: user.pic,
      token: generateToken(user._id),
    });
  } else {
    res.status(400);
    throw new Error("Error registraiton");
  }


})
```

Cyclomatic Complexity Graph

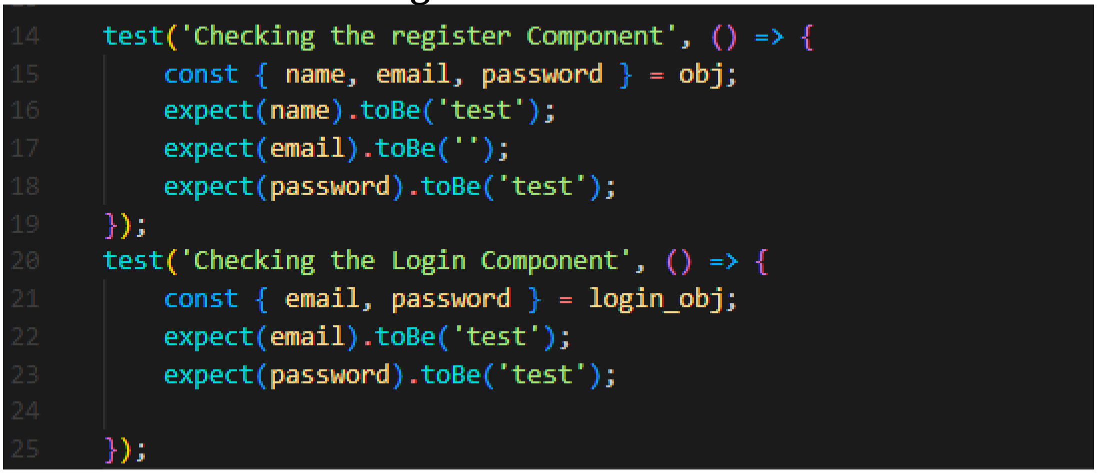


Formula = E - N + 2*P
In the above flow chart
N = 12
E = 13
P = 1
Cyclomatic complexity = 3

Test Cases:

Testing Framework used: Jest.js
Code Units Used for the unit testing are Login and Register components

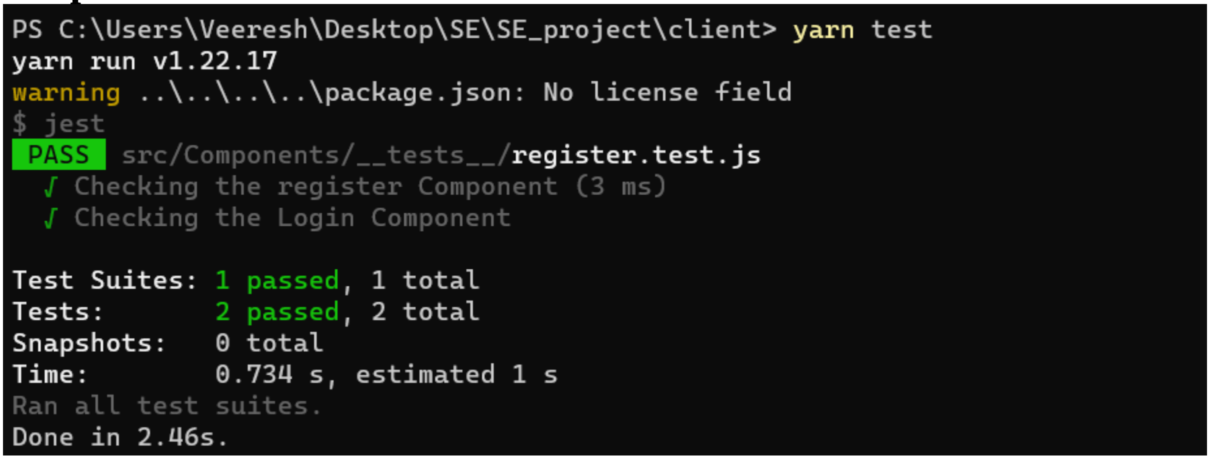
Unit test cases for testing



```
test('Checking the register Component', () => {
    const { name, email, password } = obj;
    expect(name).toBe('test');
    expect(email).toBe('');
    expect(password).toBe('test');
});
test('Checking the Login Component', () => {
    const { email, password } = login_obj;
    expect(email).toBe('test');
    expect(password).toBe('test');

});
```

Output



```
PS C:\Users\Veeresh\Desktop\SE\SE_project\client> yarn test
yarn run v1.22.17
warning ..\..\..\..\package.json: No license field
$ jest
 PASS  src/Components/__tests__/register.test.js
  √ Checking the register Component (3 ms)
  √ Checking the Login Component

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        0.734 s, estimated 1 s
Ran all test suites.
Done in 2.46s.
```