G-24 Report
CV-Maker

—

8th May, 2021

Table of Content :

Problem Statement

CV automation and resume parser CVMaker gives you the next level of resume management. It consists of CV templates which users can use to generate his/her CV. The tool also includes a parser that automatically parses the resume uploaded in some specified format, stores the parse information and helps to generate CV.

CVMaker needs to be user friendly as well as it should be able to provide admin functionalities based on the number of resumes made and number of users registering on a particular size.

Purpose:

- Our online cv maker is a simple and quick tool to create a professional CV for everyone.
- Choose one of our CV layouts or your own layout, enter your information and download your CV.
- Our WebApp focuses on helping organizations to manage large numbers of candidates who are required to make their CVs for upcoming drives.

Elicitation Techniques

- **Brainstorming**:
  - In several initial group meetings, we discussed many spontaneous ideas contributed by the members in a healthy manner.
  - We discussed what should be and should not be part of the application. i.e identifying and refining/removing ambiguous requirements.

- **Analysis of existing systems**
  - A couple of more meetings were dedicated towards analysis of existing CV-Maker systems.
  - The analysis majorly focused on tech stack used and functionality and appearance.
  - Vote based opinions were taken on parameters of each of the existing repositories/web apps.
  - Ease of use and existing problems faced by users were given priority in the voting.

Functional Requirements

- The user can parse resume formats when uploaded in the form of pdf.
- Several templates(3-4) for making resumes.
- Preview option when editing the resume.
- Functionality to save resumes in the cloud.
- Step by step guide which will help you in each important section
- Option to import previously made resume.
- Users can also provide links in their resume.
- Web applications will be able to export the document in pdf format.
- A user should be able to register an account and login into it, to store his/her information.
- A user can delete his/her account anytime.
- A user can change the password for his/her account.

Non-functional Requirements

- Security: Privacy of the user information.
- Reliability: web application should not break when a user is working.
- Performance: web application should run smoothly with less response time.
- Reusability: The last generated CVs will be stored and the user can edit it anytime in the future.
- Operating constraint : Users should have an internet connection.
- Interface constraint: UI/UX should be very minimal/unhindered as the main objective should be to easily make CVs.
- Flexibility: The website should also be responsive to run on mobile devices.
- Supported range of browsers/OS and their versions:
    - Chrome >= 45
    - Firefox >= 38
    - Edge >= 12
    - Explorer >= 10
    - iOS >= 9
    - Safari >= 9
    - Android >= 4.4
    - Opera >= 30

Technologies Used

- For Frontend we have used:
  - HTML5
  - CSS (Vanilla + Bootstrap)
  - Javascript
- For Backend We have used:
  - NodeJS
  - NPM packages such as :
    - Express
    - Mongoose
    - JWT token
    - Bcrypt
  - Mongo Atlas
- For Deployment we have used Heroku.
- For Managing our Projects we have used GitHub.

Part of Requirement not Completed

The following are the requirements we were not able to fulfill:

- Step by step guide which will help you in each important section
- Option to import previously made resume.
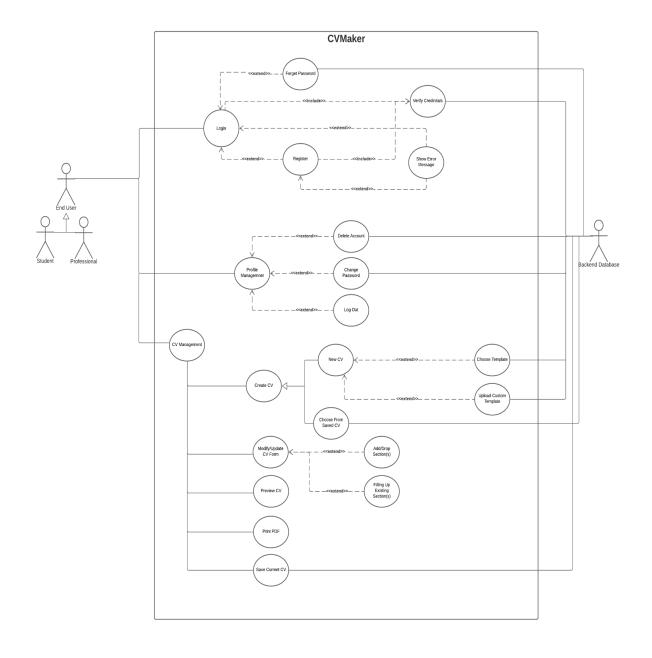- Users can also provide links in their resume.

Model

- We have Used "Incremental Waterfall Model" :
- We Divided all the requirements in 3 increments as follows:
- First in Few Meetings we decided The UI and designed a few basic components of our web app like navbars, necessary buttons, colours etc.
  - The Information regarding the same is available on the website: https://www.figma.com/file/RsIcnBCQS2qkUweMabmm6h/CVMaker?node-id=0%3A1
  - And Git Repository for the corresponding design component is: https://github.com/Aditya-Padhiar/CVMaker-Design-System

Information Regarding our Increments:

- Increment-1:
  - Authentication (local + Google OAuth).
  - Dashboard for Student and Admin.
  - My Profile Page.
  - User Schema.
  - Backend routes for authentication, Edit Profile, Admin Related Routes.
- Increment-2:

- ○ Resume Schema.
- ○ Setting Basic Template.
- ○ CV Form Page Based on chosen Template.
- ○ All action related to CV from page.
- ○ Traffic Page for Admin.
- ● Increment-3:
  - ○ Forget Password page.
  - ○ Parsing of PDF.
  - ○ Refinement of UI.

Use Case Diagram

Use Cases Description

Authentication

1)
Name: Login
Description: User can login and verification of his/her credentials will be done.
Actors: Students, Professionals.
Pre-condition: none
Flow of events
    5.1 **Basic flow**:
        1. The system validates the actor's username and password and logs him/her into the system.
        2. The system displays the dashboard and the use case ends.
    5.2 **Alternate flow**:
        1. Invalid Name / Password: If in the basic flow the system cannot find the username or the password is invalid, an error message is displayed. The actor can type in a new name or password or choose to cancel the operation, at which point the use case ends.
   6. **Goal description**: If credentials are correct then the user is redirected to dashboard or else shown an appropriate error message.
   7. **Post-condition:** User is logged in to the system

2)
    1. **Name**: Register
    2.**Description**: User can register him/her-self in the system.

3.**Actors**: Students, Professionals.
4.**Pre-condition**: none
5.**Flow of events**
     5.1 **Basic flow**:
          1.The system validates the actor's details and registers him/her into the system.
          2.The system displays the login page and the use case ends.
     5.2 **Alternate flow**:
          1.Invalid details: If in the basic flow the system finds that the username already exists or the password is invalid or password doesn't match with the confirmation password, an error message is displayed. The actor can type in new details or choose to cancel the operation, at which point the use case ends.
6. **Goal description**:
     If credentials are correct then the user is redirected to the login page or else shown an appropriate error message.
7.**Post-condition:** User is registered into the system database.

3)

     1. **Name**: Forgot-password(Change password)
     2.**Description**: User can change his/her password
     3.**Actors**: Students, Professionals.
     4.**Pre-condition**: Actors must enter the correct username and must know the correct answer to the corresponding security question.
     5.**Flow of events**
     5.1 **Basic flow**:
1. The user enters his/her username
2. Security question corresponding to the username if shown to the user.
3. User answers the security question.
4. The user can set a new password.

5.2 **Alternate flow-1**
1. The user enters his/her username
2. The username is invalid.
3. The user is shown an appropriate warning message and the use case ends.

5.3 **Alternate flow-2**
1. The user enters his/her username
2. Security question corresponding to the username if shown to the user.
3. The answer to the security question is wrong.
4. The user is shown an appropriate warning message and the use case ends.

6. **Goal description**: The user is able to change his/her password after proper authentication and the use case ends

7. **Post-condition:** If username and answer to the security question is valid then password is updated in the system database.

CV Management

4)
1. **Name**: Create CV
2. **Description**: User can choose from his/her saved cvs or select a new template.
3. **Actors**: Students, Professionals.
4. **Pre-condition**: Actors must be logged in.
5. **Flow of events**
   5.1 Basic flow:
   1. The user selects a new cv.
   2. The system opens that cv and the use case ends.
   5.2 Alternate flow:

1. The user select and already saved cv
2. The system opens that cv and the use case ends.

6. **Goal description**: The system opens selected cv and the use case ends

7. **Post-condition:** If a saved cv is opened then saved details are to be loaded, else a new cv is created in the system database.

5)
1. **Name**: Update CV
2. **Description**: User can edit sections inside the cv form.
3. **Actors**: Students, Professionals.
4. **Pre-condition**: Actors must be logged in and must have selected a cv template.
5. **Flow of events**
   5.1 Basic flow:
      1. The user edits sections inside cv form and saves it.
      2. The cv is saved and the user is shown a success message.
   5.2 Alternate flow:
      1. The user edits sections inside cv form and saves it.
      2. The cv is not saved and the user is shown an appropriate error message.

6. **Goal description**: The user is able to edit cv and the use case ends

7. **Post-condition:** If cv is saved successfully then the changes are reflected in the system database.

6)
1. **Name**: Preview CV
2. **Description**: Users can preview the cv form
3. **Actors**: Students, Professionals.

4.**Pre-condition**: Actors must be logged in and must have selected a cv template. Also at Least one field has been filled in the cv

5.**Flow of events**

5.1 **Basic flow**:

    5. The user edits sections inside cv form and saves it.

    6. The cv is saved and the user is shown a success message.

  6. **Goal description**: The user is able to preview cv and the use case ends

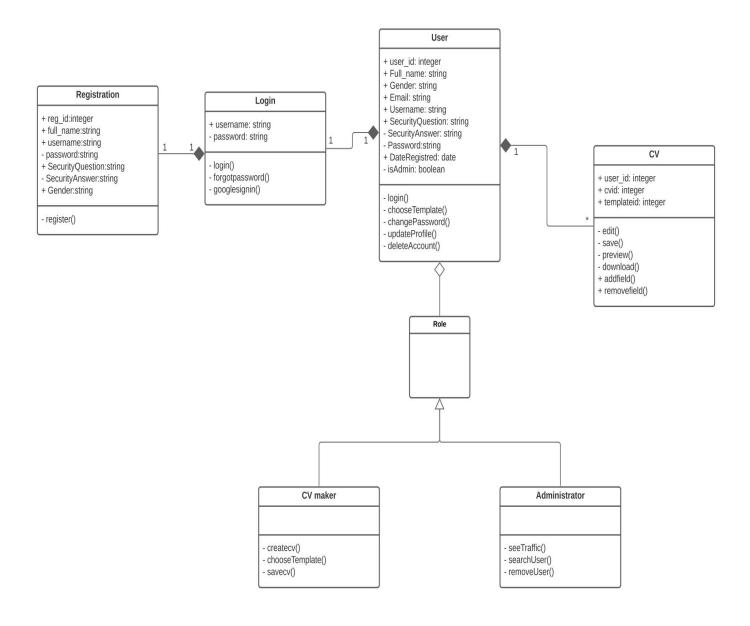 7. **Post-condition:** Edit tools are hidden in the user interface.

7)

  1. **Name**: Download CV

  2.**Description**: Users can download the cv in pdf format

  3.**Actors**: Students, Professionals.

  4.**Pre-condition**: Actors must be logged in and must have selected a cv template. Also at Least one field has been filled in the cv

  5.**Flow of events**

  5.1 **Basic flow**:

    1. The user downloads the cv.

    2. The cv is downloaded in pdf format on the user's machine

  5.2 Alternate flow:

    1. The user downloads the cv.

    2. There is some error in the download process so cv is not downloaded.

  6. **Goal description**: The user is able to download cv in his/her machine.

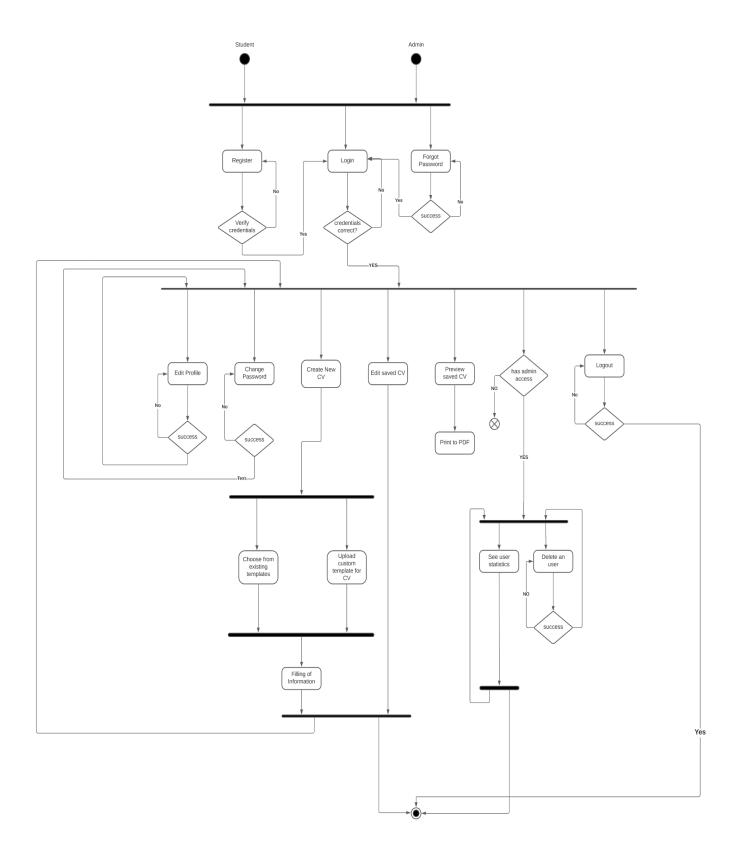 7. **Post-condition:** Download toolbar window is opened.

Profile Management

8)

        1.**Name**: Delete Account

        2.**Description**:Actors can delete their account.

        3.**Actors**: Students, Professionals.

        4.**Pre-condition**: Actors already have an account and they must be logged in.

        5.**Flow of events**

        5.1 **Basic flow**:

          1. The user deletes his/her account.

          2. Confirmation box.

         6. **Goal description**: The user is able to delete his/her account.

         7. **Post-condition:** The system deletes the account and erases the data of that account.
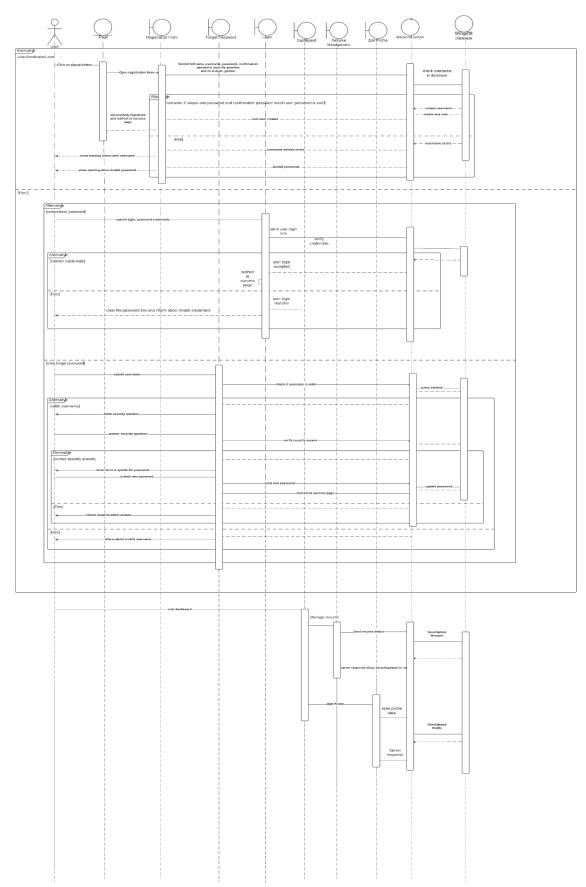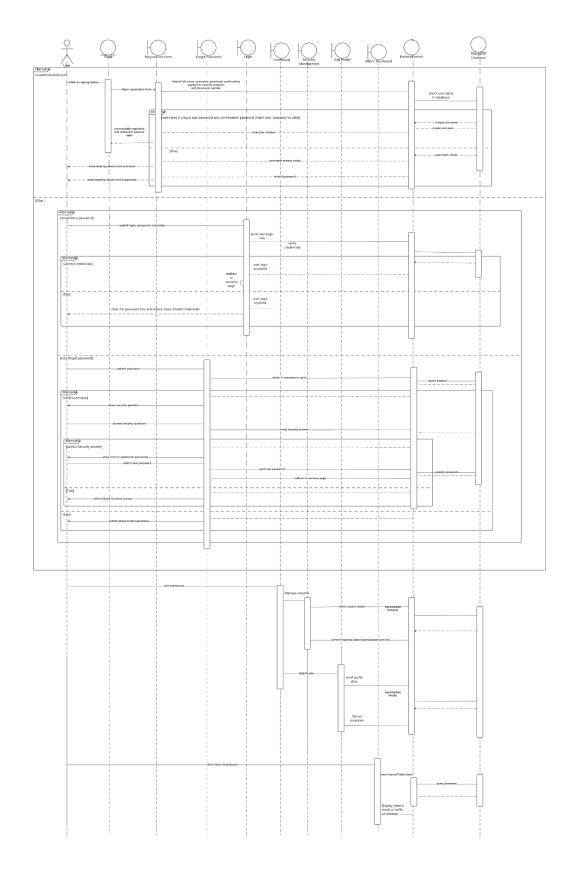
Class Diagram

Registration

+ reg_id:integer
+ full_name:string
+ username:string
- password:string
+ SecurityQuestion:string
- SecurityAnswer:string
+ Gender:string

- register()

Login

+ username: string
- password: string

- login()
- forgotpassword()
- googlesignin()

User

+ user_id: integer
+ Full_name: string
+ Gender: string
+ Email: string
+ Username: string
+ SecurityQuestion: string
- SecurityAnswer: string
- Password:string
+ DateRegistred: date
- isAdmin: boolean

- login()
- chooseTemplate()
- changePassword()
- updateProfile()
- deleteAccount()

CV

+ user_id: integer
+ cvid: integer
+ templateid: integer

- edit()
- save()
- preview()
- download()
+ addfield()
+ removefield()

Role

CV maker

- createcv()
- chooseTemplate()
- savecv()

Administrator

- seeTraffic()
- searchUser()
- removeUser()

Activity Diagram

Student   Admin

Register   Login   Forgot Password

No

Verify credentials   credentials correct?   success

Yes   No   Yes   No

YES

Edit Profile   Change Password   Create New CV   Edit saved CV   Preview saved CV   has admin access   Logout

No   No   NO   No

success   success

Print to PDF   success

Text   YES

Choose from existing templates   Upload custom template for CV   See user statistics   Delete an user

NO

Filling of Information   success

Yes

Sequence Diagram

- **Diagram for use case for regular user**
- **Diagram for use case for admin user**

- https://drive.google.com/drive/folders/1kvUYcB_2tDGJKEFoQBwMJ6Q4UxGOClxg?usp=sharing

State Diagram

Deployment Diagram

«device»
PC/Laptop

«OS»
Windows/MacOS/Linux

«execution environment»
Browser

«artifact»

«manifests»

«component»
HTML/CSS/Images/JS

User

Google OAuth API

Auth code

«device»
Google
OAuth
Server

Auth Code

Access token

«protocol»
HTTPS

«device»
Database Server

«OS»
Linux Enterprise

«execution environment»
MongoDB

«collection»
User Accounts

«collection»
Admin Account

«collection»
CV template

«protocol»
TCP/IP, Mongoose

«device»
Application Server - Heroku

«OS»
Linux Enterprise

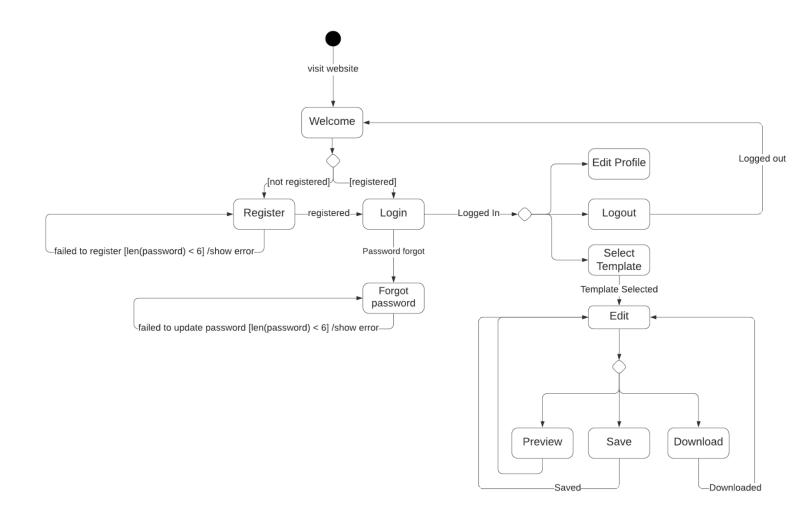«execution environment»
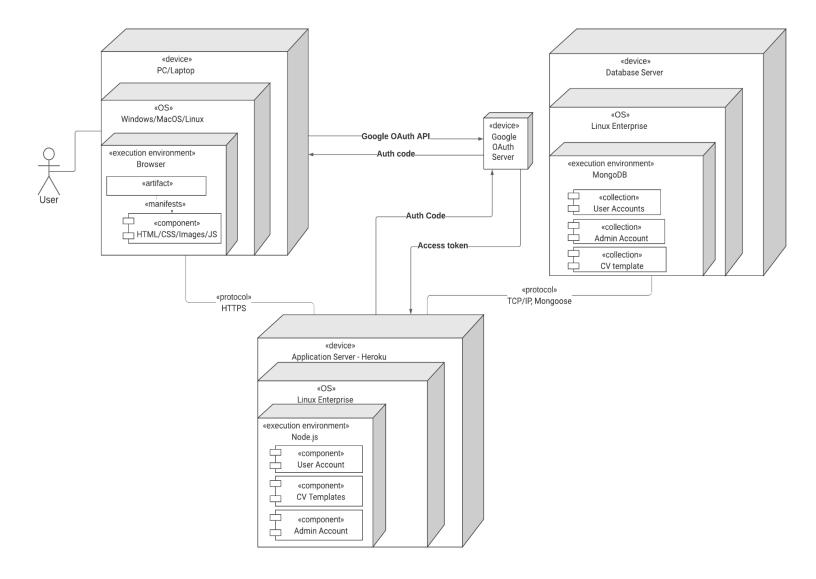Node.js

«component»
User Account

«component»
CV Templates

«component»
Admin Account

Future Work

One of the major things we can implement in the future is adding an option to add a custom template and with the help of a parser we can convert it to an HTML form and add all the necessary functionality. Also we can add more in-built templates so that users have plenty of options to choose from. Also we can add a functionality where users can create multiple resume instances from a single template.

Contribution Sheet

Aditya Padhiar (201801215):

- Frontend design in Figma
- Design system
- Dashboard page
- Classic template
- Modern template
- Profile page
- Elicitation techniques
- State diagram
- Deployment diagram

Dipankar Sugat (201801089):

- Added minimalistic template HTML CSS.
- PPT:-Wrote Achievements, key takeaways, lesson learned, Mistakes.
- Use case diagram

Harsh Modi (201801029):

- Authentication using local strategy - login, signup, security question -> RESTful routes
- Google OAuth Authentication: sign in using google account -> RESTful routes
- Backend RESTful routes for security question and password change
- Traffic page RESTful routes:

- ○ no. of new users today
- ○ no. of CVs updated today
- ○ pick rate of particular template
- ○ total user so far for last 7 days(inclusive)
- ● Merging backend with frontend(frontend JS):
  - ○ Classic-template
  - ○ Modern-template
  - ○ Minimalist template
- ● Add alert badge in forgot-password page
- ● Infra: Deployment of project on heroku
- ● SRS document: Sequence diagram of project

Himanshu Patel (201801181):

- ● Login page HTML CSS
- ● Registration page HTML CSS
- ● PPT : Achievements, key takeaways, lesson learned, Mistakes.
- ● Use-case diagram description
- ● Activity diagram

Ketul Makwana (201801084):

- ● Admin Dashboard Javascript
- ● Enhancing Admin Dashboard
- ● HTML and CSS of Traffic page
- ● Enhancing Traffic page.
- ● Demonstration video
- ● Activity Diagram
- ● Use Case Model

Krutik Parmar (201801199):

- Welcome Page
- Minimalist template
- Changing dashboard for new built-in template
- Enhancing Index page
- Elicitation techniques
- State diagram
- Deployment diagram

Mahir Ratanpara (Group Leader):

- RESTful Routes for Resume Data
- Profile Routes
- Admin Routes
- Traffic Page Routes
- Merging backend for Classic Template
- Merging backend for Modern Template
- Merging backend for Minimalistic Template
- Meeting Logs
- Maintaining Github Repository (Assignments of Issues,Linking and Merging Pull Requests)
- Contribution Tracking
- Activity Diagram
- Use Case Model

Mohit Agrawal (201801044):

- Admin Dashboard HTML and CSS

- Enhancing Admin Dashboard
- JavaScript of Traffic page
- Enhancing Traffic page.
- Activity Diagram
- Use Case Model (Diagram and description)
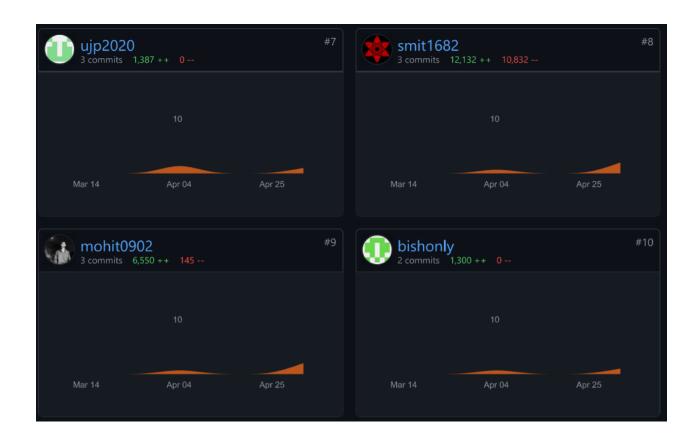
Smit Prajapati (201801122):

- Forgot Password HTML, CSS, JS
- Elicitation techniques
- State diagram
- Deployment diagram

Utsav Patel (201801148):

- Modern template
- Dashboard page
- Profile page
- Class diagram
- Use case description

Mar 7, 2021 – May 7, 2021   Contributions: Commits ▾

Contributions to main, excluding merge commits and bot accounts

MR-1301   #1
34 commits   18,612 ++   10,754 --

Harsh-Modi278   #2
19 commits   4,179 ++   649 --

ketul08   #3
4 commits   577 ++   4 --

Aditya-Padhiar   #4
4 commits   15,150 ++   11 --

341himanshu   #5
4 commits   2,769 ++   1,226 --

krutik2966   #6
3 commits   13,876 ++   305 --

Meeting Logs

27-01-2021 6:00PM - 60mins

8-02-2021 10:30PM - 120mins

10-02-2021 11:00PM - 90mins

15-02-2021 3PM - 90mins

21-02-2021 2PM - 120mins

8-03-2021 3PM - 90mins

11-03-2021 7PM - 90mins

18-03-2021 11PM - 60mins

29-03-2021 2PM - 120mins

1-04-2021 11PM - 120mins

2-04-2021 12PM - 90mins

3-04-2021 11PM - 120mins

6-04-2021 4:30PM - 90mins

20-04-2021 11PM - 45mins

21-04-2021 4PM - 150mins

5-04-2021 12PM - 240mins

6-04-2021 11AM - 240mins (with breaks)

7-04-2021 11AM - 240mins (with breaks)

Links:

https://github.com/MR-1301/G24-CVMaker

https://g24-cvmaker-prod.herokuapp.com