

CMPE - 202 Group Project Journal:

Team: Keep calm and sprint on

Ghanashri Mariyana (015231853)

Ayush Gupta (014952184)

Chetan Nain (015761122)

Viswamithra Vallabhaneni (015570516)

API Collection

<https://documenter.getpostman.com/view/11157513/UVJbHxjy>

Link to Sprint Sheet

https://docs.com/spreadsheets/d/1D519MNSB-eGfdM_dA68eonZS2Ik55_zIrgfS_2eUy3s/edit?usp=sharing

Link to Project Board

https://github.com/gopinathsjsu/team-project-keep_calm_and_sprint_on/projects

1. Tech Stack and Tools

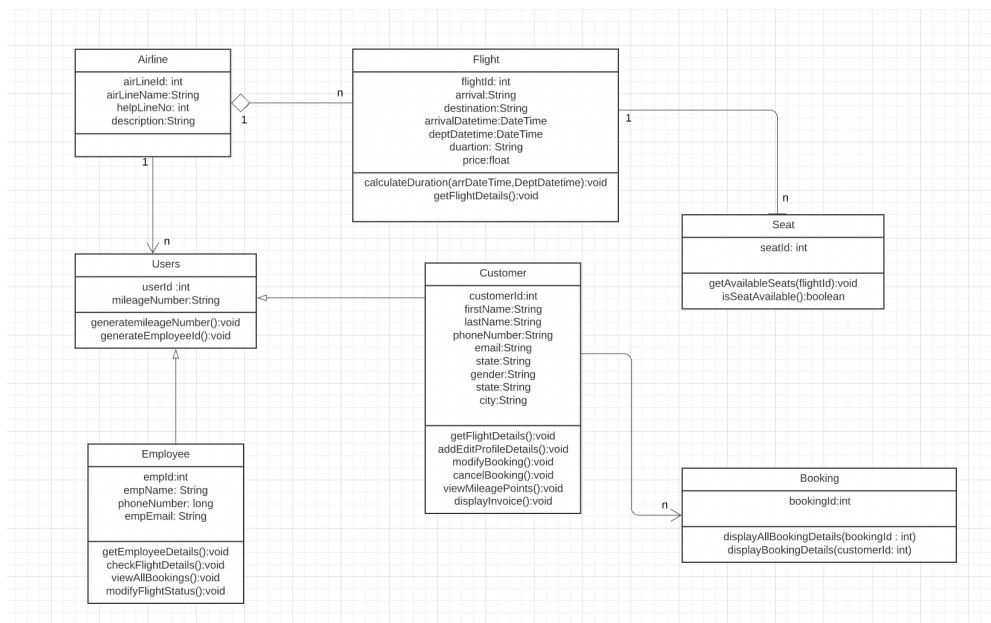
Implementation: MERN

Wireframes: Figma

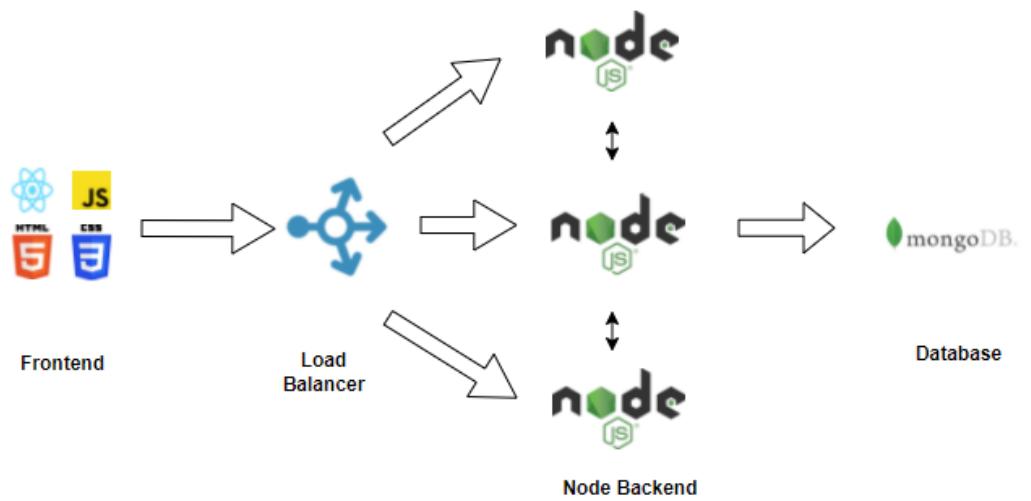
Load Balancing: AWS

Version Control: Git

2. Class Diagram



3. Architecture Diagram



4. Database Design

a. User Collection (Common for Employee and Customer)

Id	userTy pe	Empid	mileageNu mber	userna me	passw ord	phone	email	mileage Points	Address : Array	gender	location

```

const userSchema = new Schema({
  // _id: Number,
  userType: { type: String, required: true, trim: true }, // ["user", "employee"]
  empId: String, //arline_emp1
  userName: { type: String, required: true, trim: true },
  email: { type: String, required: true, trim: true },
  password: { type: String, required: true, trim: true },
  dob: {type: String },
  phone: { type: String, trim: true },
  token: {type: String, default: ''},
  address: {
    addressline1: String,
    city: String,
    state: String,
    country: String,
    zipcode: Number,
  }
})

```

```

        },
        mileageNumber: String, //airline_customer1
        mileagePoints: {type: Number, default: 0},
        gender: {type: String}
    },
{
    versionKey: false
})];

```

b. Airline Collection

AirlineId	Airlinename	HelplineNo.	Supportemail	Desc
-----------	-------------	-------------	--------------	------

```

const airlineSchema = new Schema({
    // _id: Number,
    airlineName: { type: String, required: true, trim: true },
    helplineNumber: {type: Number},
    supportEmail: { type: String, required: true, trim: true },
    desc: String,
},
{
    versionKey: false
});

```

c. FlightDetails Collection

Flight Id	Src	Dest	arrivalDateTime	deptDateTime	miles	basePrice	duration	startTime	endTime	totalAvailableeSeats
-----------	-----	------	-----------------	--------------	-------	-----------	----------	-----------	---------	----------------------

```

const flightDetailSchema = new Schema({
    // _id: Number,
    source: { type: String, required: true, trim: true },
    destination: { type: String, required: true, trim: true },
    arrivalDateTime: String,
    departureDateTime: { type: String},
    miles: Number, // distance in miles btw src and dest
    basePrice: Number, // Minimum base price to book ticket in flight
    duration: String, // In minutes or hours
});

```

```

        startTime: String,
        endTime: String,
        seats: [
            {
                seatNumber: String,
                isBooked: Boolean,
            }
        ]
    },
{
    versionKey: false
}

```

d. Bookings Collection

BookingID	userid	flightId	Seat numbers	bookingDatetim e	Total Price Paid	total Passengers	Identity Numbers	flightClass
-----------	--------	----------	--------------	------------------	------------------	------------------	------------------	-------------

```

const bookingSchema = new Schema({
    // _id: Number,
    userId: {
        type: mongoose.Schema.Types.ObjectId,
        ref: "User"
    },
    flightId: {
        type: mongoose.Schema.Types.ObjectId,
        ref: "FlightDetail"
    },
    seatNumbers: [String],
    bookingDateTime: String,
    totalPricePaid: Number,
    totalPassengers: Number,
    identityNumber: String,
    flightClass: String,
    mileagePointsUsed: { type: Number, default: 0 },
},
{
    versionKey: false
});

```

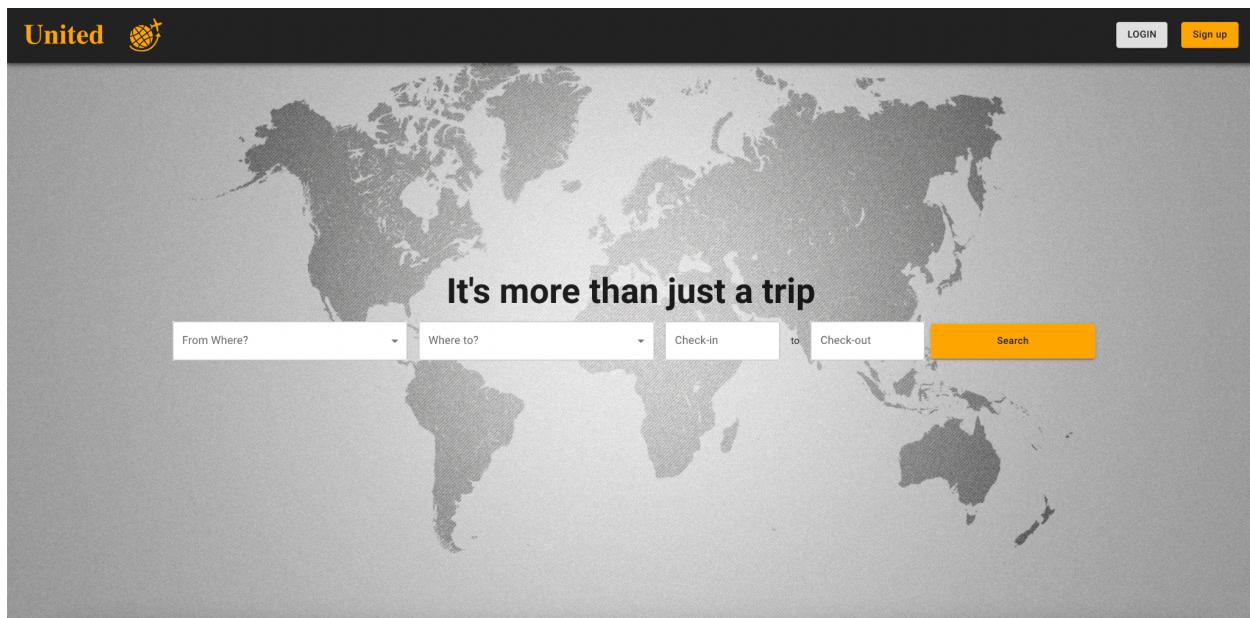
e. Seats Collection

SeatId	FlightId	SeatType (window, aisle, middle)	Class (Business/Econom y)	SeatNo.

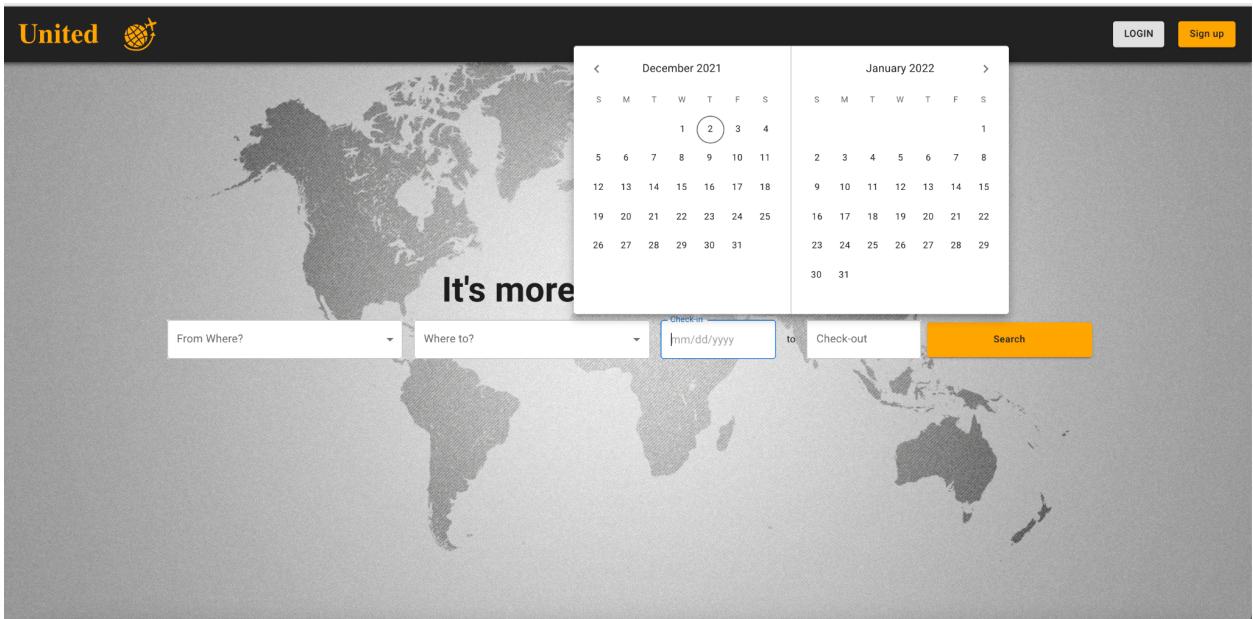
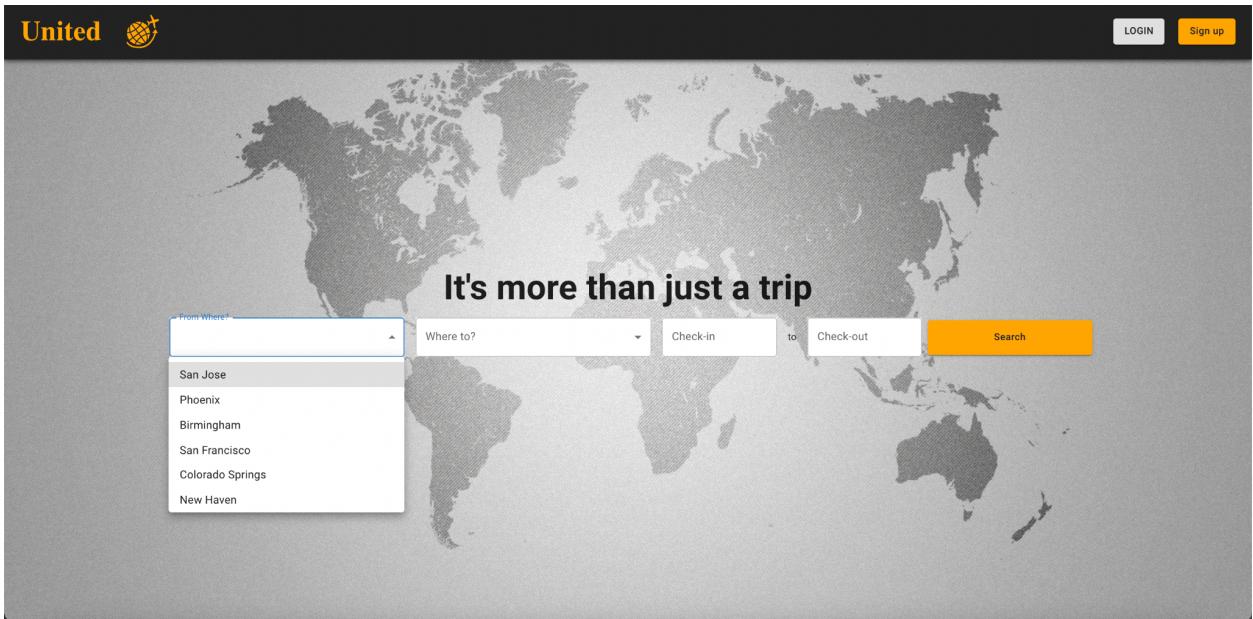
```
const seatSchema = new Schema({
  // _id: Number,
  flightId: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "FlightDetail"
  },
  seatType: String, // ["window", "aisle", "middle"]
  class: String, // ["business", "economy"]
  seatNumber: String,
},
{
  versionKey: false
});
```

5. Application Screen Shots

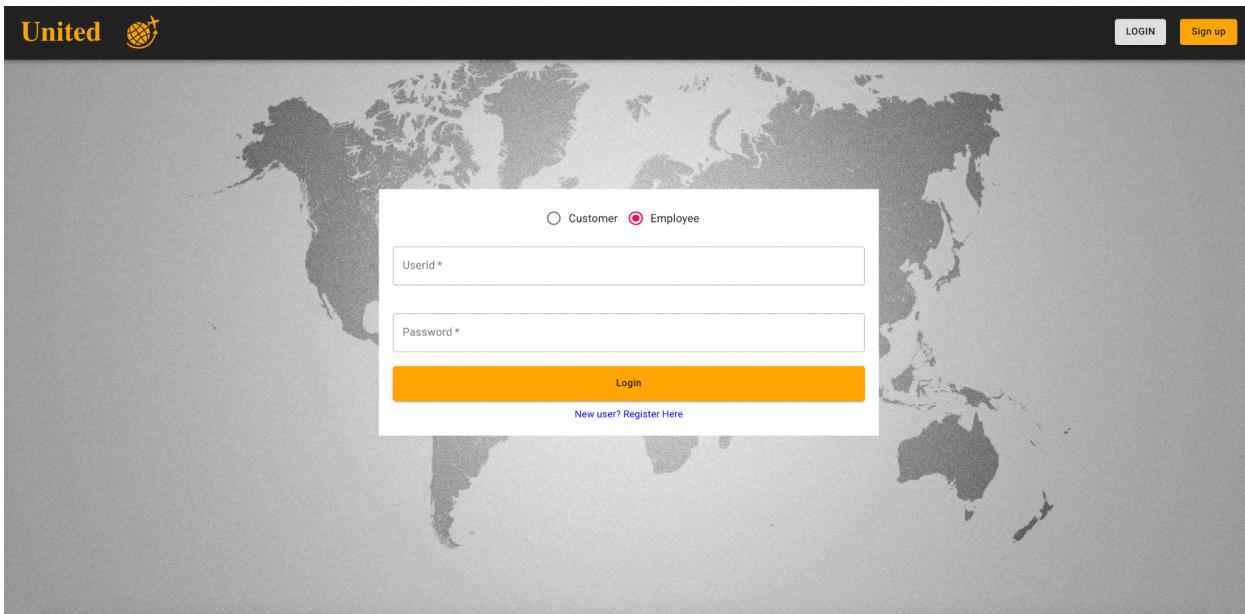
Landing Page



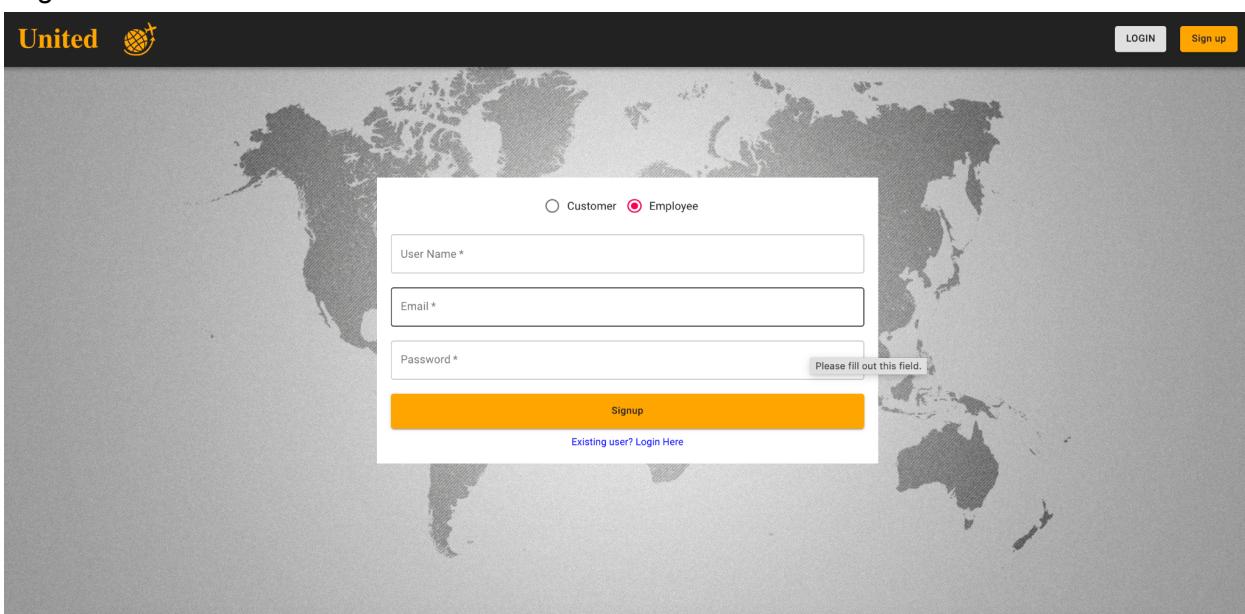
Flight Search



Login Screen



Registration Screen



Side Navbar for Employee

The screenshot shows a dark-themed web page for an employee. At the top right is the United logo. On the left is a vertical sidebar with a globe icon and four menu items: "Airline", "Profile", "Bookings", and "Logout". The main content area has a world map background. At the top, it displays "Unique Employee Identification Number: [airline_employee18](#)". Below this is a section titled "WHO WE ARE" containing text about United's route network and loyalty program. It includes a "Contact us" section with email "abc@gmail.com" and phone "16666666602", and a "Update" button.

AirLine Profile Page for Customer

The screenshot shows a dark-themed web page for a customer profile. At the top right is the United logo. On the left is a vertical sidebar with a globe icon and four menu items: "Airline", "Profile", "Bookings", and "Logout". The main content area has a world map background. At the top, it displays "Unique Customer Mileage Number: [airline_customer16](#)". Below this is a section titled "WHO WE ARE" containing text about United's route network and loyalty program. It includes a "Contact us" section with email "abc@gmail.com" and phone "16666666602".

Airline Profile Page for Employee

Unique Employee Identification Number: **airline_employee13**

WHO WE ARE

United is connecting people, uniting the world United has the most comprehensive route network among North American carriers. A modern fleet which is the most fuel-efficient among U.S. network carriers (when adjusted for cabin size). Industry-leading loyalty program that provides more opportunities to earn and redeem miles worldwide Optimal hub locations, including hubs in the four largest cities in the United States Approximately 70,000 employees reside in every U.S. state and in countries around the world

Contact us

✉️ unitedsupport@gmail.com **📞** 16693061517

Update

Unique Employee Identification Number: **airline_employee18**

Update Airline Details

Helpline Number *

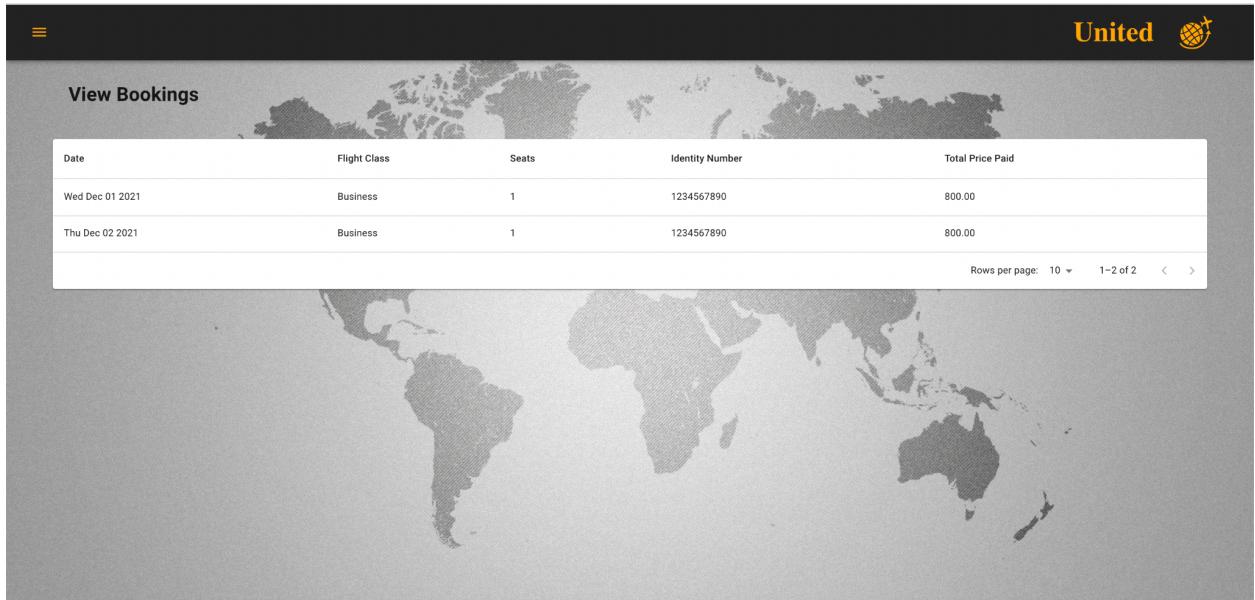
Support Email *

About *

United is connecting people, uniting the world United has the most comprehensive route network among North American carriers. A modern fleet which is the most fuel-efficient among U.S. network carriers (when adjusted for cabin size). Industry-leading loyalty program that provides more opportunities to earn and redeem miles worldwide Optimal hub locations, including hubs in the four largest cities in the United States Approximately 70,000 employees reside in every U.S. state and in countries around the world

Update

Employee Bookings Screen

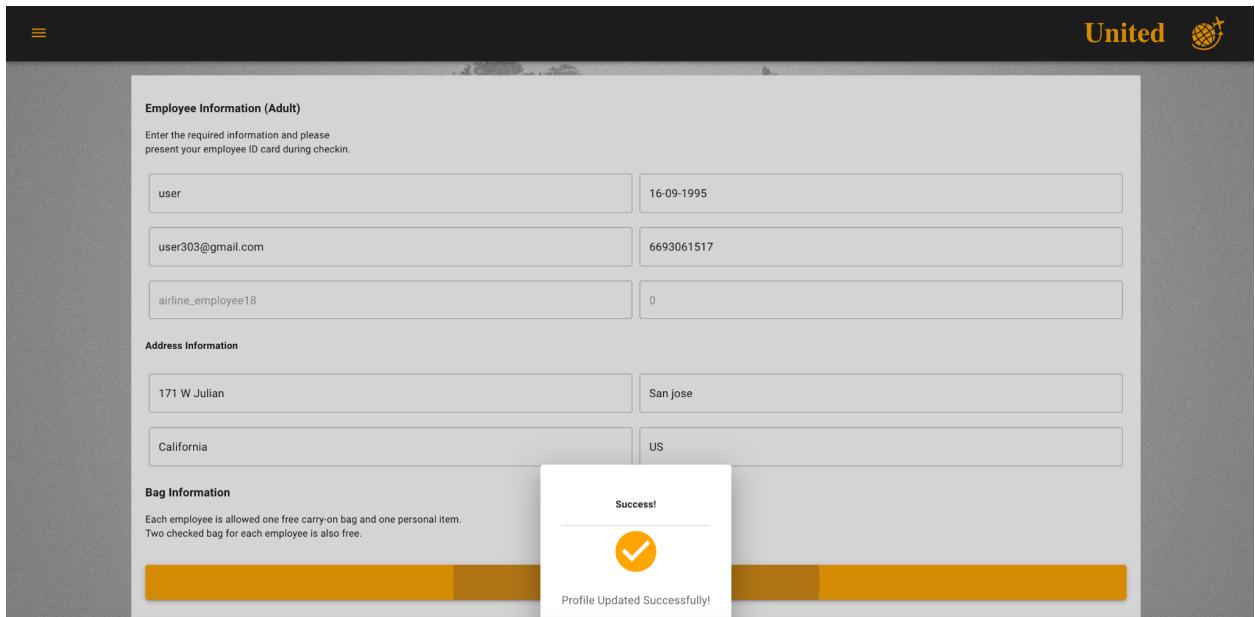


The screenshot shows the 'View Bookings' section of the United Airlines mobile application. At the top, there's a header with the United logo and a world map background. Below the header is a table showing flight bookings:

Date	Flight Class	Seats	Identity Number	Total Price Paid
Wed Dec 01 2021	Business	1	1234567890	800.00
Thu Dec 02 2021	Business	1	1234567890	800.00

At the bottom of the table, there are pagination controls: 'Rows per page: 10' and '1–2 of 2'.

Employee Profile Update



The screenshot shows the 'Employee Information (Adult)' update screen. It includes fields for Name, Date of Birth, Email, and Employee ID. Below these are address fields for Street, City, State, and Country. A 'Bag Information' section at the bottom states: 'Each employee is allowed one free carry-on bag and one personal item. Two checked bag for each employee is also free.' To the right, a success message box displays a yellow checkmark icon and the text 'Success! Profile Updated Successfully!'.

Side Navbar for Customer

The screenshot shows a dark-themed web page for United Airlines. On the left, a vertical sidebar contains a yellow globe icon and a list of navigation items:

- [View Airline](#)
- [Profile](#)
- [Flights](#)
- [View Bookings](#)
- [Logout](#)

The main content area features a banner with the text "WHO WE ARE" and a paragraph about United's global reach and employee count. Below this is a "Contact us" section with an email address and a phone number.

Customer Profile

The screenshot shows the "Passenger Information (Adult)" section of the customer profile. It includes fields for Name, Date of Birth, Email, and Social Security Number. Below this is the "Address Information" section with fields for Street Address, City, State, and Zip Code. The "Bag Information" section notes that each passenger is allowed one free carry-on bag and one personal item. At the bottom is a large orange "Update" button.

Passenger Information (Adult)

Enter the required information for each traveller and be sure it matches exactly to the government issued ID presented at the airport.

airline_customer17	09-02-1994
airline_customer17@gmail.com	6693061517
airline_customer16	0

Address Information

171 W Julian	San Jose
California	US

Bag Information

Each passenger is allowed one free carry-on bag and one personal item. First checked bag for each customer is also free. Second bag checked free is waived for loyalty program members.

Update

6. Requirement analysis:

Explored [united.com](https://www.united.com) website for a better understanding of the problem statement and came up with the following functionalities.

Personas: 2 (Employee and Customer)

- Employee:**
- a. Can modify airline Profile details,
 - b. Can add and update employee details,
 - c. Can view all the bookings,

Customer:

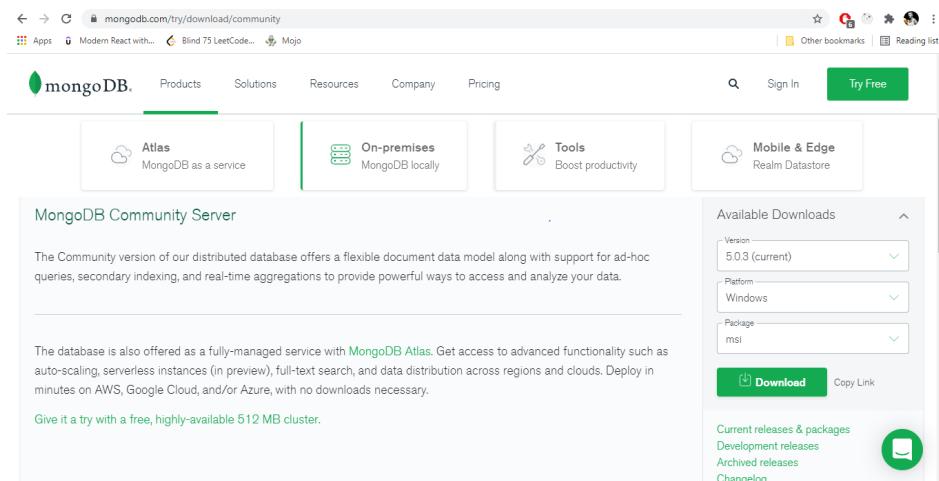
- a. Customer can view airline details,
- b. View and modify personal details,
- c. View list of flights,
- d. Search flights by destination, origin, arrival date and departure date
- e. Make bookings (Seat selection, cabin selection),
- f. Cancel or modify bookings
- g. View past bookings and invoices
- h. View Collected Mileage points
- i. Redeem Mileage points

7. Installation and Set up

MongoDB Installation:

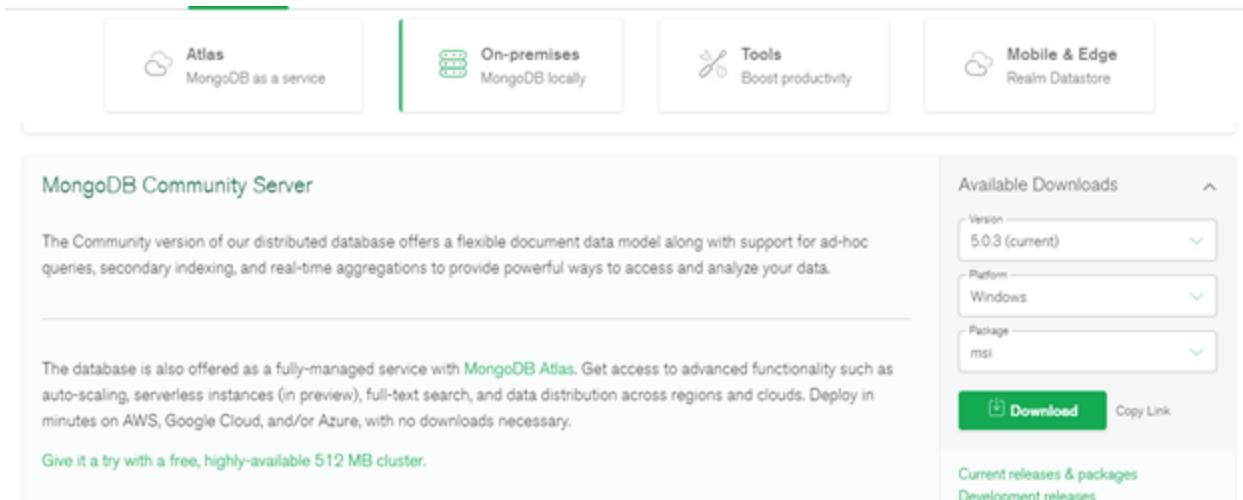
Step 1: Go on to the official website via link

<https://www.mongodb.com/try/download/community>



The screenshot shows the MongoDB website's download page for the Community Server. At the top, there are four main options: Atlas (MongoDB as a service), On-premises (MongoDB locally), Tools (Boost productivity), and Mobile & Edge (Realm Datastore). Below these, the "MongoDB Community Server" section is visible, which describes the free, open-source version of MongoDB. It includes a note about MongoDB Atlas and a link to try it for free. To the right, a sidebar titled "Available Downloads" allows users to select the version (5.0.3 current), platform (Windows), and package type (msi). A large green "Download" button is prominently displayed, along with a "Copy Link" option. At the bottom of the sidebar, there are links for "Current releases & packages", "Development releases", "Archived releases", and "Changelog".

Step 2: From the right corner choose the latest version and download the MongoDB setup.

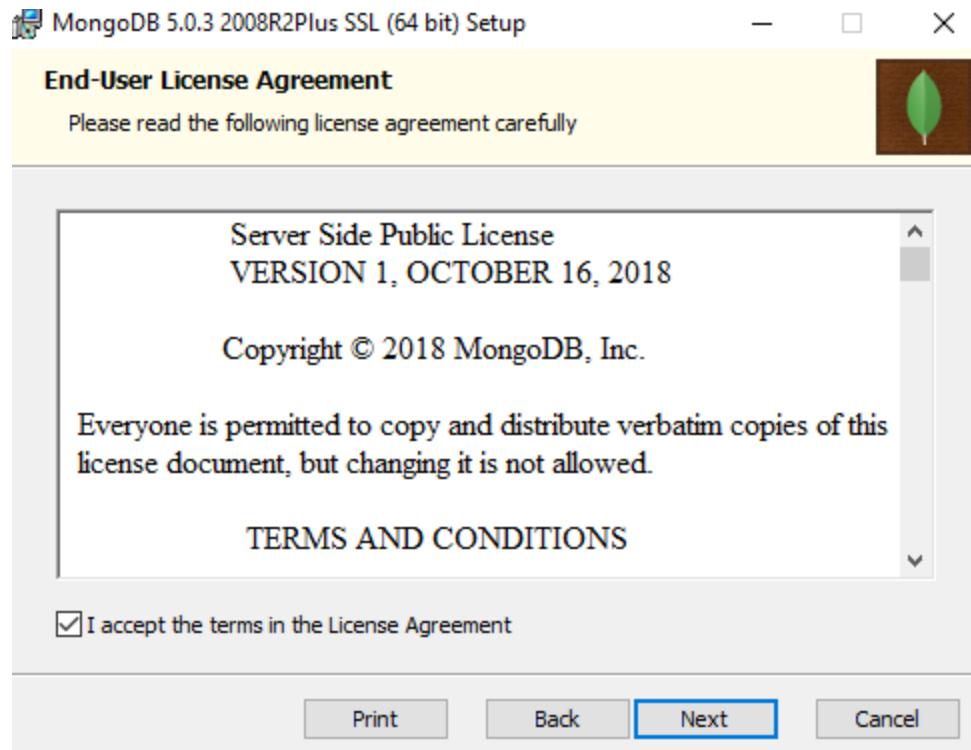


The screenshot shows the MongoDB download page. At the top, there are four navigation options: 'Atlas' (MongoDB as a service), 'On-premises' (MongoDB locally), 'Tools' (Boost productivity), and 'Mobile & Edge' (Realm Datastore). The 'On-premises' option is selected. Below this, the 'MongoDB Community Server' section is displayed, featuring a brief description of the community version and a link to 'MongoDB Atlas'. To the right, the 'Available Downloads' sidebar is shown, with dropdown menus for 'Version' (set to 5.0.3 current), 'Platform' (set to Windows), and 'Package' (set to msi). A large green 'Download' button is prominently displayed, along with a 'Copy Link' option. Below the sidebar, there are links for 'Current releases & packages' and 'Development releases'.

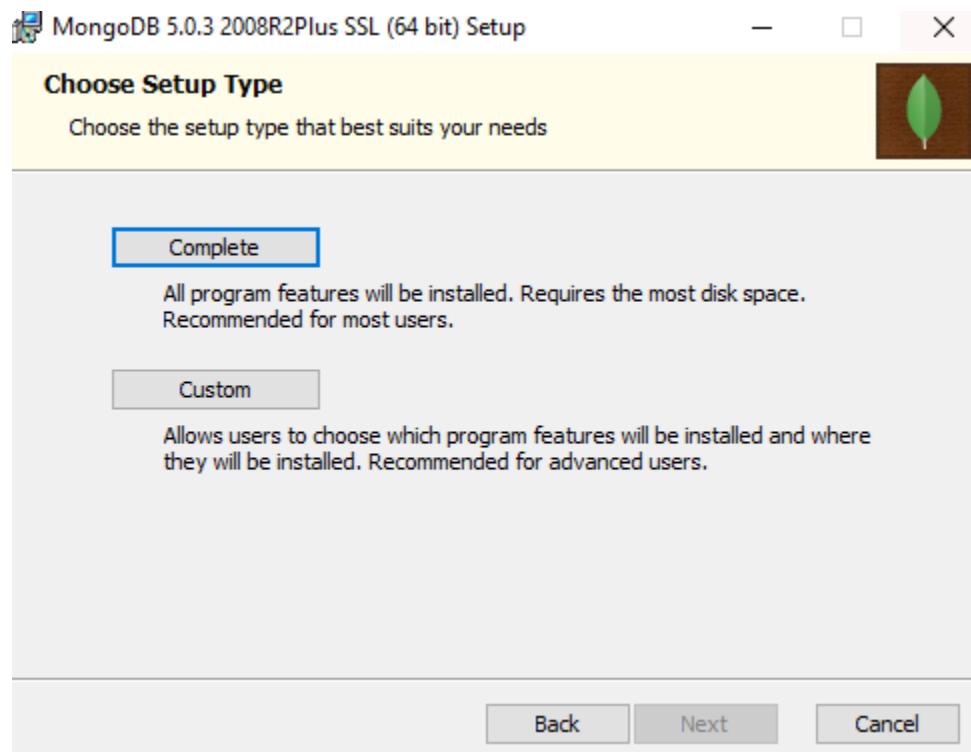
Step 3: Double click the mongoDB installed .msi file.



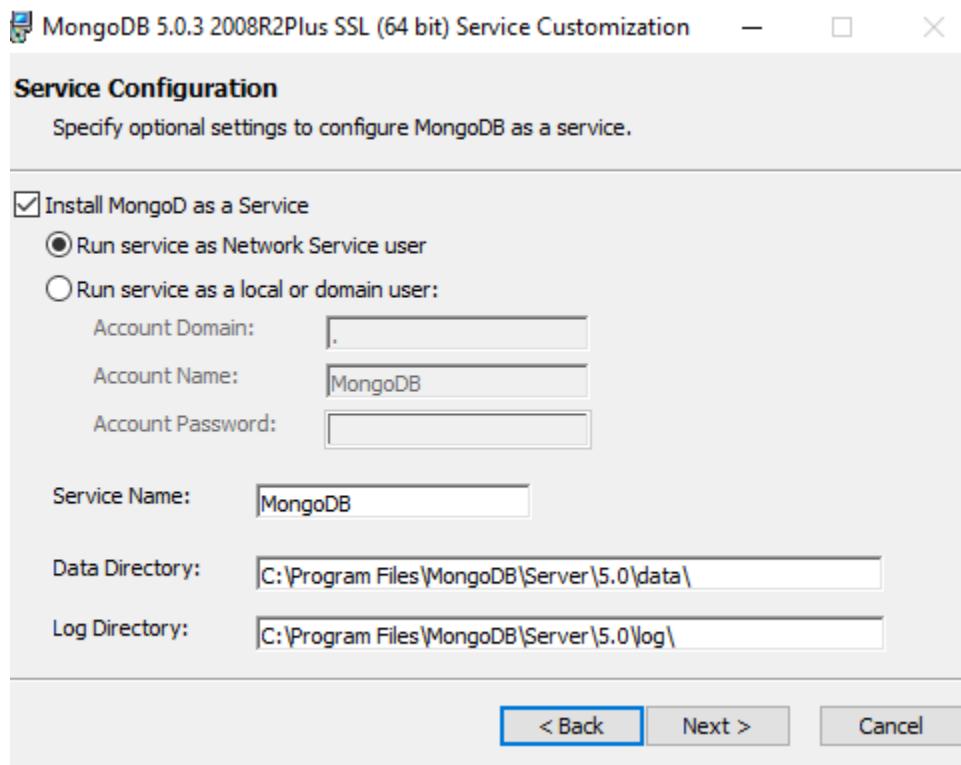
Step 4: Accept the End-User License Agreement and then click Next.



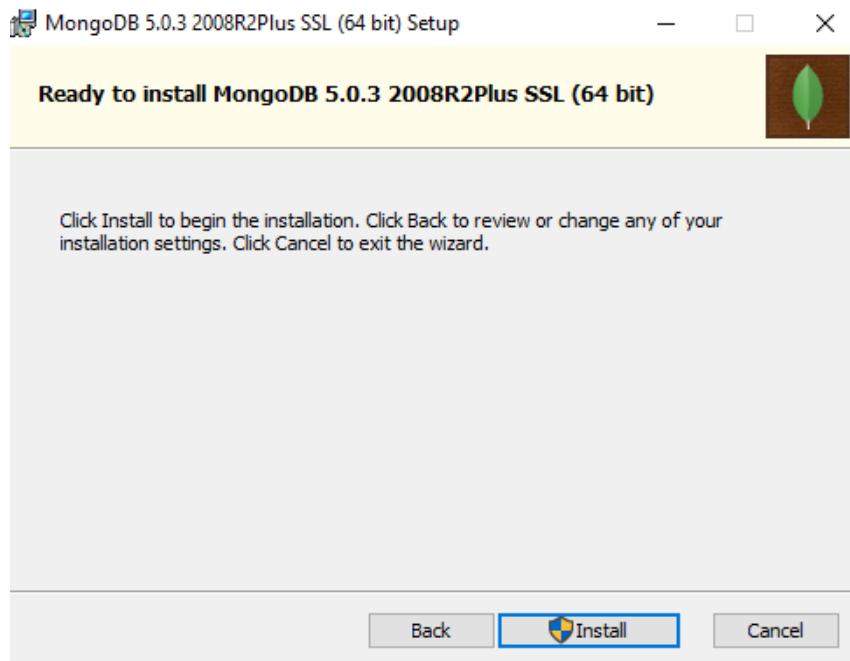
Step 5: Click on the “complete” button to install all of the components. The custom option can be used to install selective components or if you want to change the location of the installation.



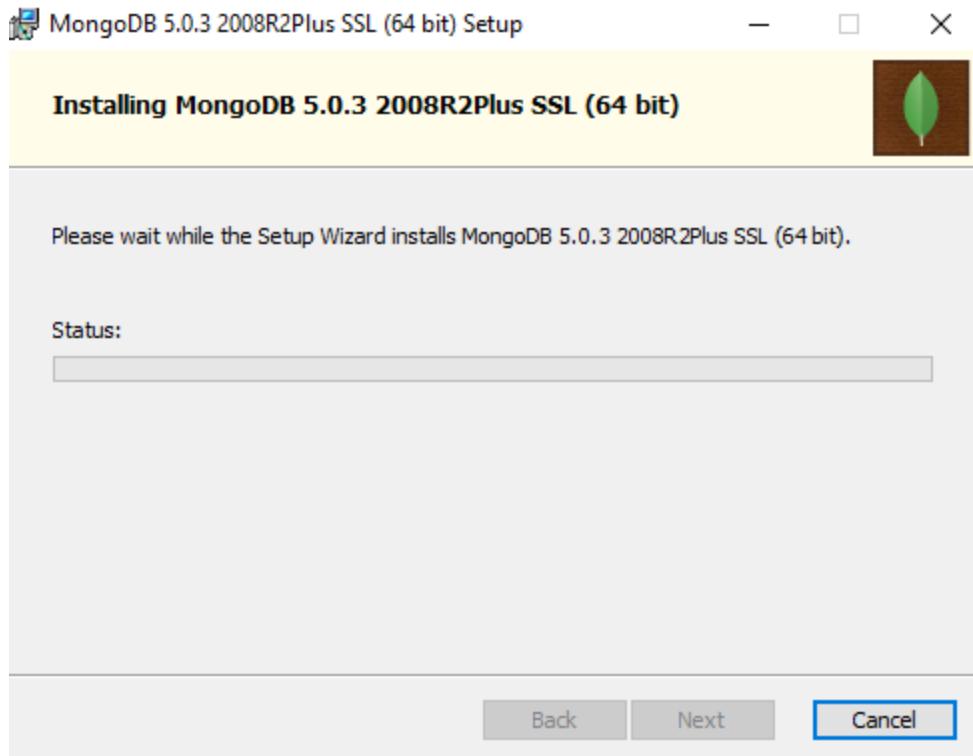
Step 6: Select “Run service as Network Service user”. make a note of the data directory, we’ll need this later and then click Next.



Step 7: Click on the install button to start the installation.



Step 8: Complete the installation by clicking on Next,



Node JS installation:

Step 1: Go on to the official website of node.js and download the appropriate version of node.

<https://nodejs.org/en/download/>

A screenshot of the Node.js official website. The header features the Node.js logo and navigation links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, CERTIFICATION, and NEWS. Below the header is a "Downloads" section. It highlights the "Latest LTS Version: 14.18.1 (includes npm 6.14.15)". It encourages users to "Download the Node.js source code or a pre-built installer for your platform, and start developing today." Two tabs are shown: "LTS" (Recommended For Most Users) and "Current" (Latest Features). Under "LTS", there are links for "Windows Installer" (with icon and file name "node-v14.18.1-x64.msi") and "macOS Installer" (with icon and file name "node-v14.18.1.pkg"). Under "Current", there is a link for "Source Code" (with icon and file name "node-v14.18.1.tar.gz"). Below these are tables for "Windows Installer (.msi)", "Windows Binary (.zip)", and "macOS Installer (.pkg)". The "Windows Installer (.msi)" table has rows for "32-bit" and "64-bit". The "Windows Binary (.zip)" table has rows for "32-bit" and "64-bit". The "macOS Installer (.pkg)" table has a single row for "64-bit".

2. Once the installer finishes downloading, double click it to launch it.

3. Click Next and you will be welcomed to the Setup Wizard.
4. Review the license agreement and click Next if you agree to the terms.
5. Choose the appropriate installation location and click Next.
6. Select the components to include while installing the node and then click Next.
7. At last, click the Install button to run the installer then on finishing it, click Finish.

Git Installation and setup:

1. Using brew install git:

```
brew install git
```

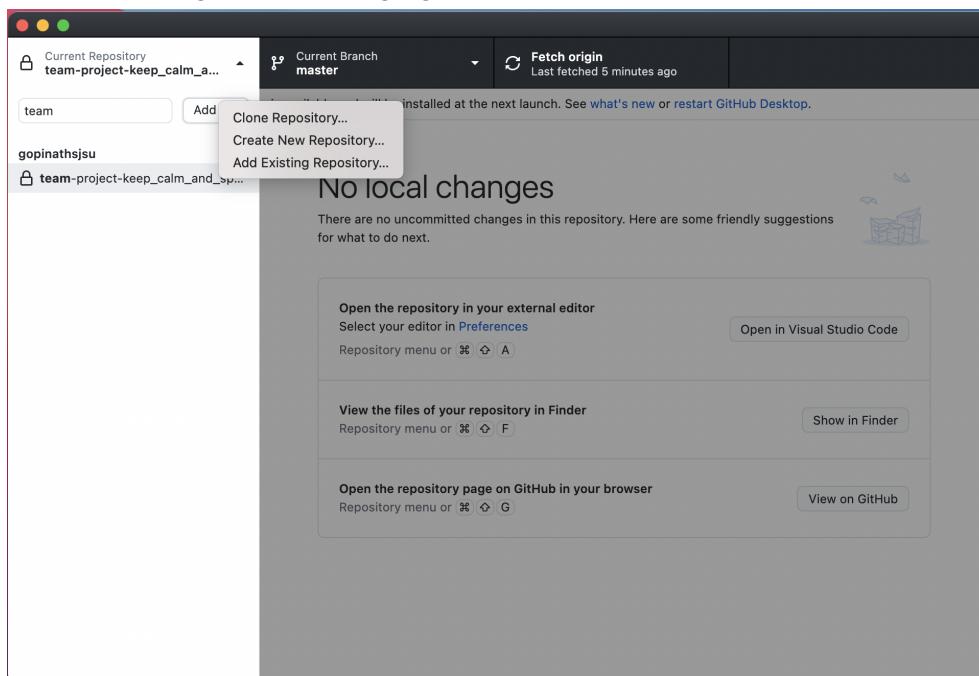
2. Check git version

```
ayushgupta@ayushs-MacBook-Air ~ % git --version  
git version 2.32.0
```

3. Install Github desktop for cli

4. Use the project repository as given and add other team members as collaborators with admin permission

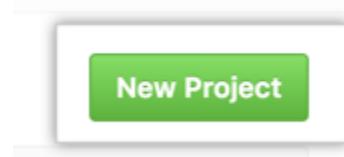
5. Take a git clone through github desktop



6. Add and commit changes and push it to repository with the github cli

Project Board Creation:

1. On the right corner of GitHub.com, click your profile icon, then click on profile.
2. In the main section on the top, click Projects.
3. Click on the “New Project” icon.



- 4.
5. Enter the project details like name and description of the project.

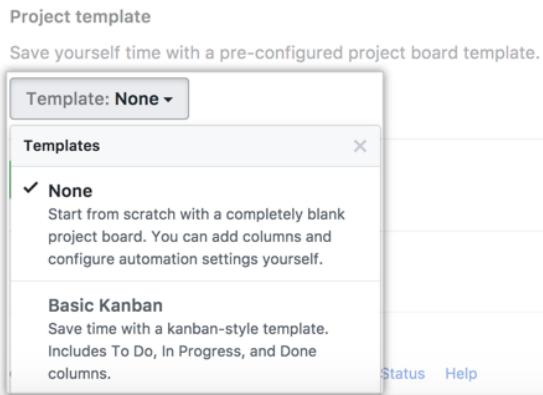
Create a new project

Coordinate, track, and update your work in one place, so projects stay transparent and on schedule.

Project board name

Description (optional)

- 6.
7. Optionally, we can use the Template from the drop-down menu.



- 8.

9. According to our requirements we can choose the visibility.

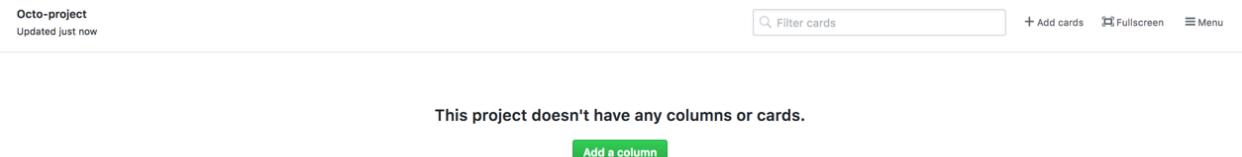
Public
Anyone on the internet can see this project. You choose who can make changes.

Private
You choose who can see and make changes to this project.

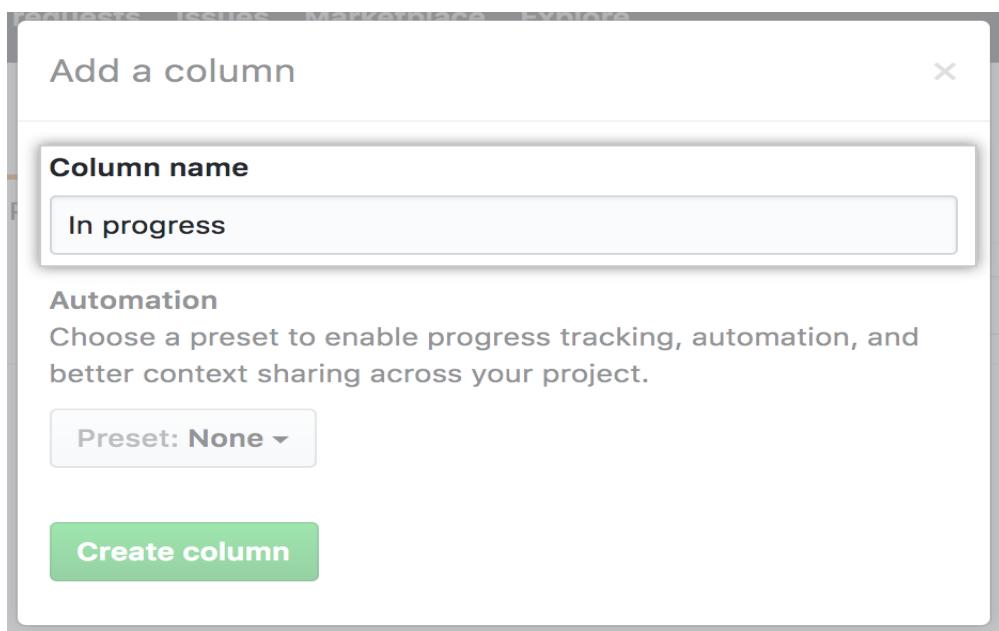
10.

11. Click on the Create project button.

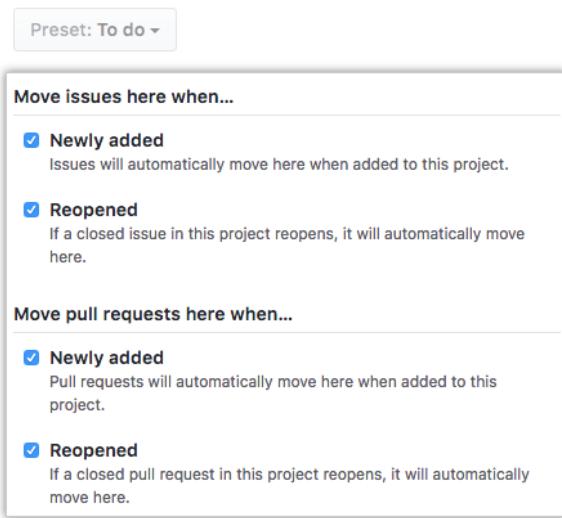
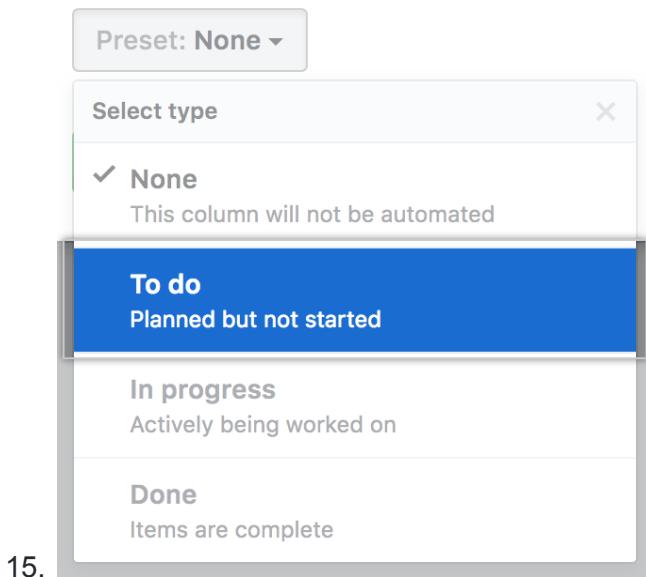
12. On the project board we can add a column by clicking the Add a column option.



13.



14.

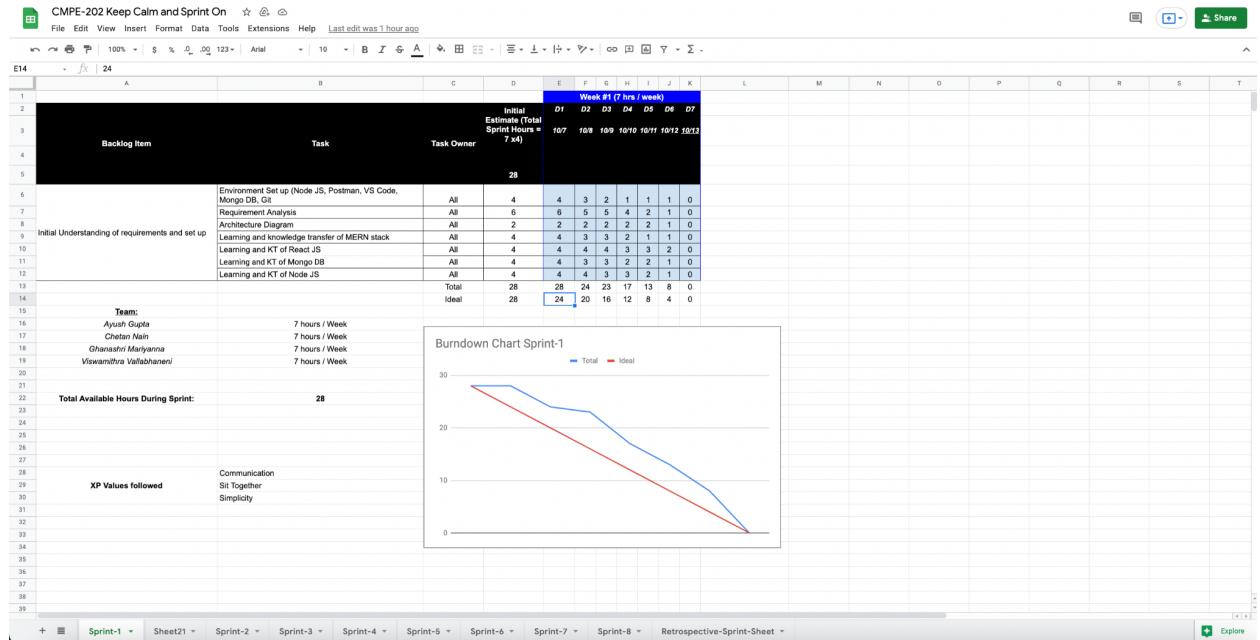


17. On Clicking the column we can add columns that are in accordance with our requirements.

Sprint Sheet Creation:

A sprint sheet is used to keep track of all work done by the team members and each team member is assigned a total of 7 hours per week for each sprint.

Below is the detailed sprint 1 task sheet.



The above screenshot details all the tasks accomplished and the hours spent by each team member on each task during this sprint.

8. Application pages and flow

Landing page (Sign up, login, search flight functionality) - Same for Employee and Customer.

Employees will login with **employee id**, provided on first sign up.

Customers will login with the **mileage number** provided on first sign up.

Customer:

Customer Dashboard (Where customers can search for the flights).

Flight details page (Where customers can see the selected flight's trip timings for the selected date),

Booking page (Where customers can select the seat type, seat numbers, cabin type etc and redeem mileage points)

Booking Confirmation Page - (Must be navigable here from history page)

Booking history page (Where customers can view their past flight bookings and invoices),

Personal details page (Where customers can update their personal details, location etc)

Employee:

The Airline Profile page will have the airline details (Which is read only to customers). Can be modified by the employee.

The Booking page will have the list of bookings and customer details in each booking.

The Employee Profile page will house all the details of the logged in employee. The employee can update the profile details.

9. Wireframes

<https://www.figma.com/file/X1TCtQ3gBogsE7f3hTJgxg/Untitled?node-id=17%3A5347>