
Implementation Document

for

CAB SHARING

Version <1.1>

Prepared by

Group 18:

Group Name: Code Closed

AAKASH LAWLA
ABHISHEK GURJAR
AKASH KUMAR BHOI
RISHABH MUKATI
PRINCE KUMAR AHIRWAR
HARIOM SHAKYAWAL
GOPAL AGGARWAL
UJJAWAL GOYAL
SOURABH MINA
SOURAV ANAND

190006
190037
190082
190704
190646
190354
200390
201058
200996
200997

aaklawala@iitk.ac.in
gurjara96@gmail.com
akashbhoi525@gmail.com
mukatiirishabh02@gmail.com
princeprinceahirwar@gmail.com
hariomkoli306@gmail.com
gopalagggarwal5858@gmail.com
ujjawalgo247@gmail.com
sourabh2002wow@gmail.com
souravanand1982@gmail.com

Course: CS253

Mentor TA: Pinaki Chakraborty

Date: 20 March, 2022



CONTENTS	II
REVISIONS	II
1 INTRODUCTION	1
2 UNIT TESTING	2
3 INTEGRATION TESTING	3
4 SYSTEM TESTING	4
5 CONCLUTION	5
APPENDIX A - GROUP LOG	6

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1 Implementation Details

Provide the details of programming languages, frameworks, libraries, database systems, build systems, etc. that you have used for implementing your software.

Provide a brief justification of choosing any tool by stating its benefits over the alternatives.

We have used mainly basic web development techniques.

Programming Languages:- HTML5, CSS3, JavaScript

Frameworks:- express.js, Bootstrap, Mongoose

Runtime environments:- Node.js

Node.js middleware:- express, mongoose, body-parser, passport, email-validator, validate-date, bcryptjs

Database Systems:- MongoDB

For the frontend of our web application we have used HTML, CSS, JavaScript, Bootstrap5, font-awesome (for star icons in feedback page).

For the back end of our web application we have used JavaScript Frameworks like Node.js, express.js, mongoose, mongodb.

For the database management we have used mongoDB.

(We have used bcrypt for password hashing. We have used passport for authentication instead of jwt because we can implement jwt in passport and can swap the authentication mechanism without affecting other parts of our application. We have used email validator and date validator to validate email and date, because email should be valid, if not then anyone can write anything in email address and can affect our website and also so one person can have one account with one email. As it's easy to use and it has no restriction on how many items can be in a document unlike mySQL. We can have different numbers of items in different documents in the same collection. And on top of that we can write our code in JavaScript, no need to learn other commands. That's why we chose to use mongodb over mySQL. We used mongoose and express.js as they make it easy to connect with the mongodb database. We have used bootstrap for styling our website but to change it according to us we also used CSS, when it required.)

2 Codebase

Provide the link to your github repository.

Mention briefly how to navigate the codebase.

Github Link : <https://github.com/Akashbhoi/Cabsharing>

assets - Our website designing related files are stored under this directory, and are further broken down into several categories. are stored under this directory, and are further broken down into several categories.

assets/css - stores files for styling pages and stores images that are used in our website also.

Config - Contains the link to our MongoDB database in <key.js>, and also holds the passport authentication of the system <passport.js>, i.e. identification in the login errors <login.js>.

Docs* - Currently holds uploaded Design and SRS documents *(subject to update).

models - our websites all commonly used models were stored in models (user, driver, feedback, notification, travel) folder.

Node_Modules - Holds the modules used in the code, (with instruction via readme files) on how the modules are being used.

routes - It contains all the backend support for requests like welcome, login, logout, profile, dashboard, driverdetail, travel form etc. of our website pages in .js format.

views - We get all frontend files written in .ejs files for all pages in the views folder.

views/welcome - This is the home page of our website which will be shown to the user before login.

views/dashboard - This is the home page of our website which will be shown to the user after login.

views/dashboarddriver - This is the home page of our website which will be shown to the driver after login.

views/profile - This is a page where the user can see his/her details.

views/driverprofile - This is a page where the drivers can see his/her details.

views/login - This is a page where a user/driver can sign in using his/her username and password or can sign up if not registered.

view/register - This is a page for registration of new users/new drivers.

view/feedback - In this page users can rate on various components and can write feedback on textbox.

views/driverdetail - This is a page where the drivers can see his/her details.

views/search - The search page allows the user to search for a ride to a particular destination.

views/travelform - The page is to be filled by the user to apply for the cab by providing the details of the journey.

views/about - The about page holds the details of the team members on the project.

views/request - The page shows the requests for a ride partner for our ride.

views/contact - This is the page for the user to contact the developer, in case he comes across a difficulty using the software.

3 Completeness

Provide the details of the part of the SRS that have been completed in the implementation.

Provide the future development plan by listing down the features that will be added in the (may be hypothetical) future versions.

Current Work

Following are the parts that have been completed in the implementation :

- **Registration** : We have successfully made the registration page for both driver and campus junta.
The registration page has following functionalities:
 - Users can register them as a student or as a driver.
 - Users have to enter their details like Name, Address, Email, Phone Number etc.
 - If the user's Email ID is found unregistered, then the registration process will continue, else the user will be redirected to the Login Page.
 - Users have to pick a username and set a password.
 - Username has to be an unique Username and Password has a minimum of 8 characters including both case letters and numbers and a special symbol and max 16 characters.
 - After successful registration the user will be redirected to the login page.
- **Login** : We have successfully made the login page for both driver and campus junta.
The login page has following functionalities:
 - Users have to enter the username and password.
 - If login credentials are found valid then they enter the main page.
 - Otherwise It will show a warning message and a forgot password option.
- **Profile**: We have successfully made the profile page for our users..
The profile page has following functionalities:
 - Users will be able to see their details.
 - They will be able to update their details.
 - They will be able to Logout and after logout it will redirect to the login page.
- **Update Profile** : We have provided an option to our users for changing their profiles..
The update profile option has following functionalities:
 - Users have to enter new details like Phone number, Email etc.
 - If a user wants to update the email id or mobile no., it must be unregistered.

- **Travel detail and Match** : We have successfully made the function for travel details and matches for our users..

The function has following functionalities:

- Users will select a date and time and fill in the required information like starting location and final destination.
 - They will get the list of the available people based on their input information.
 - They will have to send a request to the most matched person.
- **Search** : We have successfully added the search feature.

The search feature has following functionalities:

- Users can search for matches related to his\her interests.
- They can filter the result according to their needs.
- Users can send a request for cab sharing to the suitable person.

- **Notification** : We have successfully added the notification feature.

The notification feature has following functionalities:

- Users will get a notification if their request will be accepted.
- They will also receive information about the matched person.

- **Requests** : We have successfully added the request feature.

The request feature has following functionalities:

- Users can accept or reject request which have been sent to them and

- **Feedback** : We have successfully added the page for taking feedback from users.

This page has the following functionalities:

- Users will be provided options to rate our primary services..
- Users can also give suggestions in the dialogue box provided.

- **About** : We have successfully added the about page in our platform.

The about page has the following information:

- Users will get the contact details of developers.

- **Driver Details** : We have successfully provided features for providing details of the driver.

This feature has the following functionality :

- Users will find the details of drivers and their contacts.
- Users will find the details of rates offered per kilometer by drivers.
- Users will find the details of availability and booking status of the driver.

Future Plans

The features which we are planning to include in our software are:

- **Chat :** We will add the chat feature in our software.
The chat feature will have the following functionalities:
 - Users get an option to chat with their matches.
 - They can send and receive messages from his\her partner.

- **Verification Link :** We will add a function to send the verification link of email for various tasks.
This function will have the following functionalities:
 - The email will be verified when the user tries to reset the password with the help of Forgot Password utility.
 - The email will be verified when the user tries to change the default email address in the user profile.

Appendix A - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to implement your software>

DATE	TIME	Discussion
8th March	5.00 pm - 5.20 pm	Tell every member to read the Implementation document and decide to meet on 9th March at 6.00 pm.
9th March	6.00 pm - 6.40 pm	In zoom meet, discuss Implementation among us, clear some doubts, and discuss unclear doubts with TA on discord. Also divide work between us.
13th March	3.00 pm - 3.45 pm	Evaluate work among us and help each other where anybody is stuck.
15th March	5.00 pm - 6.15 pm	Discussion on zoom meetings and getting feedback from each other and discussing Implementation.
17th March	3.00 pm - 4.30 pm	Evaluate work among us and help each other where anybody is stuck.
19th March	5.00 pm - 5.45 pm	Finalise Implementation and document and make log entry, fix minor errors.
20th March	8.00 pm -8.30 pm	Document reviewed by TA and make changes as he suggests.