Carpool App

Project Document

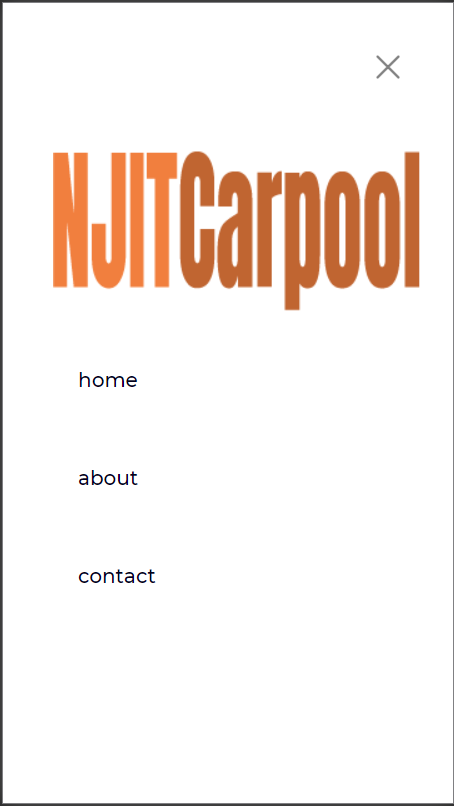
Professor Guilin Wang

CS656

Group 1

Kevin Anton, Soumik Chaudhuri, Kefin Sajan

Carpool App

Have you ever wondered if you could save money on gas and have thought about playing your part in saving our planet? 

Well we have a solution to that problem. By introducing this application to the students at NJIT, we aim to play a part in reducing pollution, saving students on commute costs, and reducing traffic on our roads. Our objective is to allow NJIT students to exchange information and schedule a carpool to/from NJIT. We hope to connect people through their smartphones to each other. In this document we will go through how we designed and implemented our idea.

As a small group with a lack of experience in mobile app development, we faced a plethora of issues. We started designing the application defining the objectives and how we should go forward. Focusing on Android application development, we faced problems as none of the team members knew kotlin and lack of experience in a new programming language created a blockage in progress. As some of the members knew python, we proceeded in researching python libraries such as Kivy and BeeWare. These were open-source tools that would convert a python script into Android apk file to be easily installed. Although Kivy was open sourced, it used proprietary user interaction elements such as labels buttons and elements of a UI toolkit which made it difficult for novice programmers to learn the whole library. The other option was BeeWare which used the native android toolkit but support was lacking to jump right in and create a mobile application in a couple of weeks.

After countless hours of research and meetings, we decided to create an applet using Hermit Lite Apps Browser which we will refer to as just Hermit. Hermit converts web sites into lite apps that are tightly integrated with Android. One of Hermit’s claims is that it can create an app based off of any website. Other browsers such as Chrome can turn a website into an app as well but not with the functionality that Hermit can provide. Hermit integrates lite apps into the android OS and allows the user to search from within the app, receive notifications from Atom and RSS feeds, and customize privacy, content, and behavior settings for every single lite app individually. 

Hermit offers many features including a content blocker. Hermit’s included content blocker can block ads, malware, fake news, and any other unwanted content and is enabled by default for created lite apps. Hermit automatically downloads a block list that contains hundreds of thousands of hosts known to be associated with advertising, malware, fake news, and other unwanted content and automatically blocks any requests from those domains. Although ad blocking is a cat and mouse game and the block lists need to be updated regularly. Hermit’s automatic updates include updates to the block lists. Some ads are loaded from popular domains and are not detected by Hermit’s content blocker when looking at network traffic alone. Ads can still make their way past the content blocker due to unforeseen circumstances and Hermit will add ads to their block list if they are sent a screenshot with the exact URL. Hermit allows the user to also pick and choose which block lists they want enabled or disabled.

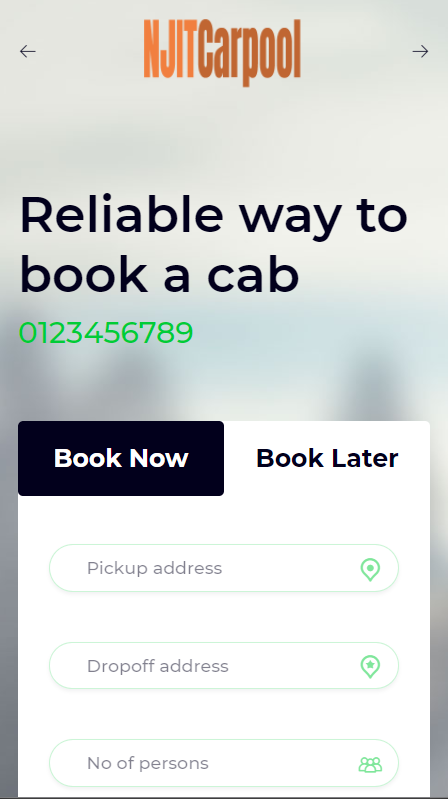
Hermit’s content and malware blockers are built-in and do not require a plugin unlike other browsers. This allows Hermit to process faster web browsing since it automatically blocks most ads, javascript, images, or animations. This can also save data usage which may be important to mobile users. Hermit’s ad blocker block-list is available publicly from Hermit’s source code repository. A second list containing phishing and malware sites is available on Hermit’s malware protection provider’s servers. Hermit also keeps the user’s browsing information private. When a suspicious site is visited, Hermit requests a double-check to ensure the site is still considered a risk and uploads partial information derived from the address back to Hermit’s malware servers. It is a similar process to other open-source browsers such as Firefox.

Hermit values privacy highly and utilizes incognito mode by default for all links where there are no existing apps. Hermit can also be used as an android’s default browser. Hermit also has options for handling external links inside a lite app. These options include opening the link inside hermit or using the system’s default browser. Internal links are automatically opened by Hermit inside the lite app. Link handling is also a configurable feature inside Hermit.

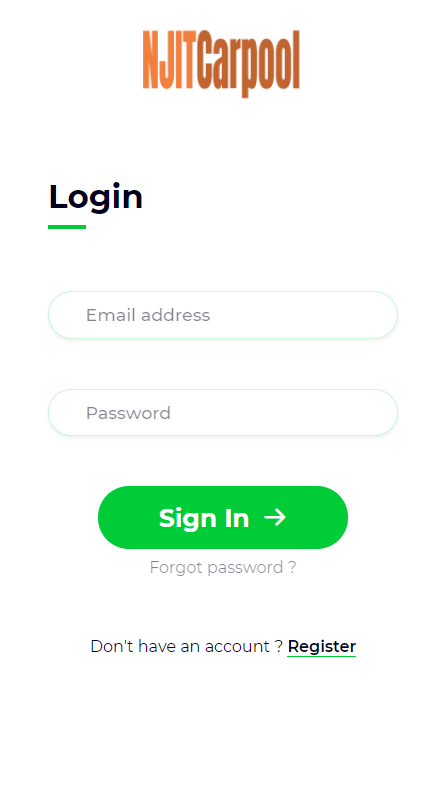
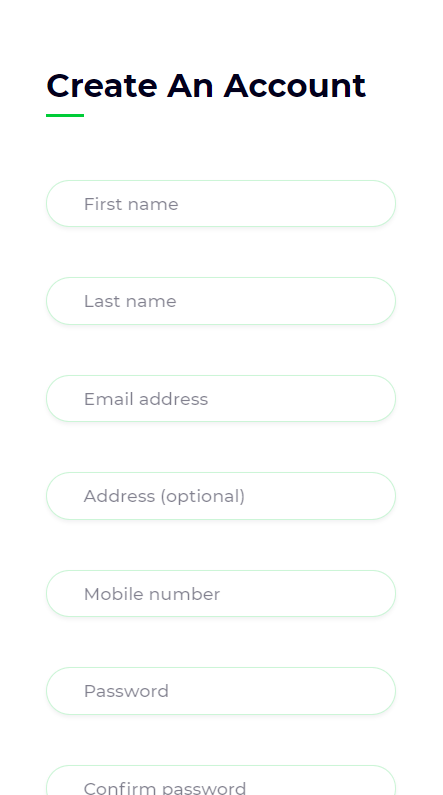
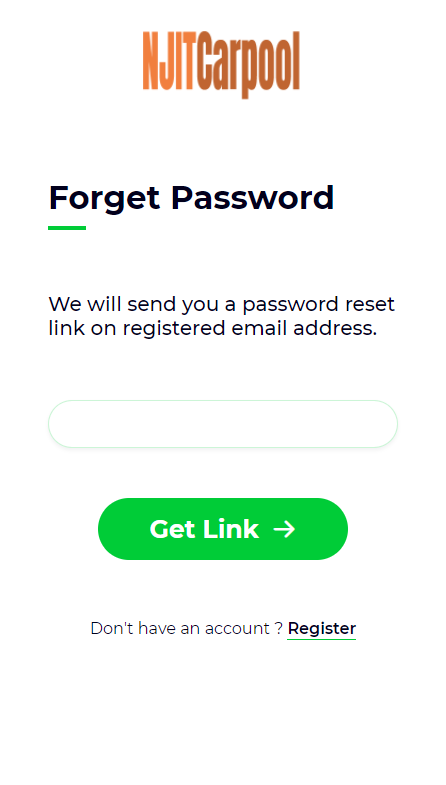
Privacy is highly valued by Hermit as the browser also offers Do Not Track technology that enables users to opt out of tracking by websites they do not directly visit which may include analytic services, advertising networks, and social platforms. Do Not Track technology is not perfect, user friendly, nor comprehensive. Do Not Track is a relatively new web standard being supported by Hermit with the goal of privacy in mind.

Malicious sites might also want to directly open links that might mess up an android device but Hermit safeguards against links to Google Play Store and other external sites with a warning to anything that tries to share to another app, open in another app, or install any other app. Lite apps are very light as well in terms of size with most sizes of lite apps being under five megabytes each. Performance is quite excellent as well due to the extra privacy of Hermit which cuts off instances of javascript and options to enable/disable images.

Hermit is a fantastic tool for creating lite apps in android. For our purposes, the conversion to a lite app was made very simplistic using Hermit. All that was needed is the main URL of the web site that holds the NJIT carpool infrastructure which would then be used to create a lite app. After that, we were able to modify many settings of the newly created lite app to our own specifications. The default name of the app is ‘Fleet Frontend’ and has limited features and a slick design.



Users can expect to see a login screen and then a page where users can book a ride either now or later. The required user input fields are pickup address, drop off address, number of persons, preferred vehicle type. There is also an additional comment field for any specifications outside the normal fields. On that same page we also have a simple display of currently only two models of vehicles that are offered by the app. Selection is very limited, but one can browse the selection nonetheless. Under that would be the services provided advertisement which displays the services that the completed app would offer in addition to carpooling such as twenty-four hour support, home pickups, and easy booking. Further underneath that would be the testimonials from customers after they leave a review from further implementation. Testimonials include a face, a name, and a review. The very bottom of site pages includes useful links, contact us information, and a little about us.

Our user login requires an email address from the client and a password. To create an account, a client has to provide their first name, last name, email address, mobile number and password. We have built a feature to reset the client’s password if it has been forgotten. The client can provide their registered email address and we will provide a link to reset the password.

While we strive for perfection, our team which includes Kevin Anton, Soumik Chaudhuri and Kefin Sajan has shown what is possible when peers gather and work on a common goal. We are constantly updating and working to improve our initial creation.