SE PROJECT 2
PACKTRAVEL

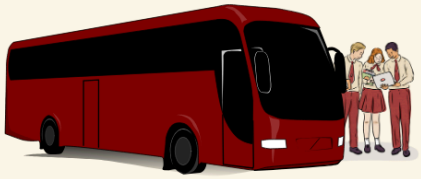
Members: Group 29
Suyash Sanjay Pustake - spustak
Shruti Vivek Kulkarni - skulkar6
Sharwari Amol Akre - sakre

Github link: https://github.com/TripleS-org/PackTravel_G29

Poster:

PackTravel 2.0: Enhancing Your Carpooling Experience

The Future of Shared Travel



PackTravel is an innovative web application designed to connect individuals looking to share rides, reduce travel costs, and enhance the commuting experience. Our mission is to foster a community that prioritizes cost-effective and eco-friendly travel options.

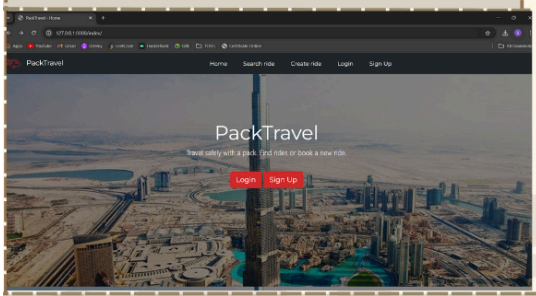
SCAN ME

TECH STACK

CSS, HTML5, Python, JS, Bootstrap, MongoDB

TEST CASE ADDITIONS

1. Verify that attempts to log in with incorrect credentials return an appropriate error message.
2. Ensure users can log out successfully and are redirected to the correct page afterward.
3. Verify that users can rate and review rides, and that this feedback is stored and displayed correctly.



OBJECTIVES

REDUCE TRAVEL COSTS
PackTravel aims to help users lower their commuting expenses by promoting shared rides. Whether it's carpooling with personal vehicles, sharing a taxi, or organizing a bus ride, the application makes it easier for travelers to connect with others, cutting down on individual transportation costs while promoting eco-friendly travel.

IMPROVE RIDE COORDINATION
he app simplifies the process of organizing and joining shared rides by offering features like ride creation, ride requests, and a dedicated forum for each trip. This helps users manage logistics and communicate effectively with others in the ride group, ensuring smooth coordination for both drivers and passengers.

LEVERAGE TECHNOLOGY FOR ROUTE AND FARE OPTIMIZATION
Integration with Google Maps allows users to view routes, distance, and duration of trips, making it easier to plan commutes. Additionally, the machine learning-powered cab fare estimation tool helps users get accurate fare predictions based on time and date, enabling more informed decisions when selecting transportation options.

ENHANCE USER EXPERIENCE
The project is designed to make the overall user experience more intuitive and streamlined. Users can quickly create rides, send requests, accept or decline passengers, and receive notifications when their ride reaches full capacity. The app is tailored to reduce friction in ride management, ensuring that both ride owners and riders enjoy a seamless experience.

ENABLE SCALABILITY
PackTravel is built with scalability in mind. By designing the backend as stateless RESTful APIs and implementing horizontal scaling with MongoDB, the system can handle increasing loads of data and users. Integrating features like content delivery networks (CDNs) and message queues ensures that PackTravel can scale efficiently as the user base grows, offering faster load times and better system performance.

METHODOLOGY

- **PLANNING AND SYSTEM ARCHITECTURE**
  The initial phase focused on identifying user needs through market research and surveys to create a ride-sharing solution. The architecture was designed for scalability, utilizing a microservices approach with RESTful APIs for seamless communication between components. MongoDB was chosen for its flexibility and ability to handle large data sets, while the Google Maps API provides critical route visualization and distance calculations.

- **FRONTEND DEVELOPMENT**
  The user interface was developed with a focus on usability and responsiveness. Using HTML, CSS, JavaScript, and Bootstrap, the design allows for quick navigation and easy access to features such as ride creation, joining rides, and searching for routes. User experience was prioritized by implementing intuitive controls, an engaging layout, and smooth animations to enhance interaction with the application.

- **BACKEND AND DATA HANDLING**
  Django served as the backbone for the backend services, handling user authentication, ride logistics, and database interactions. It manages complex business logic while ensuring secure access to user data. Additionally, machine learning algorithms were implemented to analyze historical ride data, enabling accurate fare predictions for cab rides based on various factors like time and date, thereby enhancing user decision-making.

- **TESTING AND CONTINUOUS FEEDBACK**
  The application underwent rigorous testing, with over 130 test cases designed to validate core functionalities and ensure system stability. This phase included unit tests, integration tests, and user acceptance tests. Post-deployment, the team established feedback loops to gather user insights, allowing for ongoing improvements and the development of new features for subsequent versions, such as ride merging and enhanced security measures like two-factor authentication.

KEY UPGRADES IN CURRENT PROJECT

- **ENHANCED RIDE CREATION OPTIONS**
  In the latest version of PackTravel, users are no longer limited to creating a single route for each ride. The platform now supports the ability to create multiple routes, giving users the flexibility to mix and match between bus, cab, and personal vehicle options. This enhancement significantly improves the user experience, making it easier for travelers to plan trips that accommodate their schedules and preferences. Whether users need to switch between transportation modes or find alternate paths to reach a destination, this upgrade ensures that PackTravel can handle complex, multi-route journeys seamlessly.

- **ADVANCED AUTOCOMPLETE FOR SOURCE AND DESTINATION**
  The new version of PackTravel includes an improved autocomplete feature that speeds up the process of entering source and destination points. Drawing from enhanced location data, this feature now makes it easier than ever for users to accurately select their starting and ending locations with minimal effort. The smart autocomplete suggests locations as users type, offering real-time suggestions that are not only faster but also more relevant to the travel routes users tend to take. This upgrade greatly enhances usability, particularly for those unfamiliar with precise addresses or locations, making ride creation smoother and more efficient.

- **MACHINE LEARNING-DRIVEN FARE ESTIMATION**
  One of the most exciting upgrades in PackTravel is the integration of machine learning for cab fare estimation. By analyzing past travel data, ride patterns, and external factors like date and time, the system now provides highly accurate fare predictions for cab rides. This new feature helps users make more informed decisions when selecting travel options, ensuring they can budget their rides more effectively. The machine learning model continually improves with new data, meaning the predictions will only get more precise over time, offering a significant advantage over static fare charts.

- **REAL-TIME NOTIFICATIONS**
  Managing group rides is now more convenient with the addition of real-time email notifications for ride owners. Whenever a ride reaches full capacity, the system automatically sends out an email to the ride owner, ensuring they stay informed without needing to check the status of their ride manually. This automation saves time and reduces the chance of communication lapses, enabling ride owners to focus on other trip logistics. It also creates a seamless experience for users trying to join rides, as the owner can promptly respond to new requests or adjust capacity as needed.

- **USER PREFERANCES**
  Enable users to create and manage their profiles with personal information and travel preferences. This allows users to see their likes and select rides and co-passengers accordingly, enhancing the ride-matching experience.

- **FEEDBACK SYSTEM**
  Create a feedback mechanism that allows users to leave comments on rides after completion. This provides valuable insights into ride quality and encourages continuous improvement of the service.

FUTURE SCOPE

- **MACHINE LEARNING FOR RIDE OPTIMIZATION**
  By leveraging historical data, the system can predict ride demand based on factors like time of day and location, leading to improved user satisfaction and optimized earnings for ride owners through dynamic pricing.

- **RIDE HISTORY**
  By rendering the data collected from the user after a feedback is submitted after a ride is over, functionality of the user being able to see their previous rides can be seen.

- **NOTIFICATION UPDATE**
  Users can see updates about their bookinng and ride status if notifications about bookings accepted etc are given.

GROUP MEMBERS: 1. SUYASH PUSTAKE 2.SHRUTI KULKARNI 3.SHARWARI AKRE

No.	Notes	Self Assessment	evidence
1.	Workload is spread over the whole team (one team member is often Xtimes more productive than the others…	3	-https://github.com/TripleS-org/PackTravel_G29/pulls -https://github.com/TripleS-org/PackTravel_G29/issues?q=is%3Aissue+is%3Aclosed
2	but nevertheless, here is a track record that everyone is contributing a lot)	3	-https://github.com/VSangarya/PackTravel/compare/main...TripleS-org:PackTravel_G29:G29
3	Number of commits	3	-https://github.com/VSangarya/PackTravel/compare/main...TripleS-org:PackTravel_G29:G29
4	Number of commits: by different people	3	-https://github.com/VSangarya/PackTravel/compare/main...TripleS-org:PackTravel_G29:G29
5	Issues reports: there are many	2	-https://github.com/TripleS-org/PackTravel_G29/issues
6	Issues are being closed	3	-https://github.com/VSangarya/PackTravel/compare/main...TripleS-org:PackTravel_G29:G29
7	Docs: doco generated, format not ugly	3	-https://github.com/TripleS-org/PackTravel_G29/blob/G29/README.md
8	Docs: what: point descriptions of each class/function (in isolation)	3	-https://github.com/TripleS-org/PackTravel_G29/blob/G29/README.md
9	Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	3	-https://github.com/TripleS-org/PackTravel_G29/blob/G29/README.md
10	Docs: why: docs tell a story, motivate the whole thing, deliver a punchline that makes you want to rush out and use the thing	2	-https://github.com/TripleS-org/PackTravel_G29/blob/G29/README.md
11	Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.	3	-https://github.com/TripleS-org/PackTravel_G29/blob/G29/README.md
12	Use of version control tools	2	=https://github.com/TripleS-org/PackTravel_G29/tree/G29
13	Test cases exist	3	https://github.com/TripleS-org/PackTravel_G29/tree/G29_tests

14	Test cases are routinely executed	2	https://github.com/TripleS-org/PackTravel_G29/tree/G29_tests
15	Issues are discussed before they are closed	3	https://github.com/TripleS-org/PackTravel_G29/issues
16	Chat channel: exists	3	-https://github.com/TripleS-org/PackTravel_G29/pull/4
17	Test cases: a large proportion of the issues related to handling failing cases.	3	https://github.com/TripleS-org/PackTravel_G29/tree/G29_tests
18	Evidence that the whole team is using the same tools: everyone can get to all tools and files	3	https://github.com/TripleS-org/PackTravel_G29
19	Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	3	https://github.com/TripleS-org/PackTravel_G29
20	Evidence that the whole team is using the same tools (e.g. tutor can ask anyone to share screen, they demonstrate the system running on their computer)	3	https://github.com/TripleS-org/PackTravel_G29
21	Evidence that the members of the team are working across multiple places in the code base	3	-https://github.com/TripleS-org/PackTravel_G29/commits/G29/
22	Short release cycles	3	-https://github.com/TripleS-org/PackTravel_G29/commits/G29/
23	The file .gitignore lists what files should not be saved to the repo. See [examples]i(https://github.com/github/gitignore)	3	-https://github.com/TripleS-org/PackTravel_G29/blob/main/.gitignore
24	The file INSTALL.md lists how to install the code	3	-https://github.com/TripleS-org/PackTravel_G29/blob/G29/INSTALL.md
25	The file LICENSE.md lists rules of usage for this repo	3	-https://github.com/TripleS-org/PackTravel_G29/blob/G29/LICENSE
26	The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example	3	-https://github.com/TripleS-org/PackTravel_G29/blob/G29/CODE-OF-CONDUCT.md
27	The file CONTRIBUTING.md lists coding standards and lots of tips on how to extend the system without screwing things up; e.g. see example	3	-https://github.com/TripleS-org/PackTravel_G29/blob/main/CONTRIBUTING.md

28	The file README.md contains all the following - Video -DOI badge: exists. To get a Digitial Object Indentifier, regiser the project at Zenodo. DOI badges look like this: Zenodo doi badge -Badges showing your style checkers - Badges showing your code formatters. - Badges showing your syntax checkers. - Badges showing your code coverage tools -Badges showing any other Other automated analysis tools -	3	-https://github.com/TripleS-org/PackTravel_G29/blob/G29/README.md

Notes	Answer	Evidence
Does your website and documentation provide a clear, high-level overview of your software?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does your website and documentation clearly describe the type of user who should use your software?	Yes	https://github.com/TripleS-org/PackTravel_G29
Do you publish case studies to show how your software has been used by yourself and others?	No	
Is the name of your project/software unique?	Yes	https://github.com/TripleS-org/PackTravel_G29
Is your project/software name free from trademark violations?	Yes	https://github.com/TripleS-org/PackTravel_G29

Is your software available as a package that can be deployed without building it?	Yes	https://github.com/TripleS-org/PackTravel_G29
Is your software available for free?	Yes	https://github.com/TripleS-org/PackTravel_G29
Is your source code publicly available to download, either as a downloadable bundle or via access to a source code repository?	Yes	https://github.com/TripleS-org/PackTravel_G29
Is your software hosted in an established, third-party repository likeGitHub (https://github.com), BitBucket (https://bitbucket.org),LaunchPad (https://launchpad.net) orSourceForge (https://sourceforge.net)?	Yes	https://github.com/TripleS-org/PackTravel_G29
Is your documentation clearly available on your website or within your software?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does your documentation include a "quick start" guide, that provides a short overview of how to use your software with some basic examples of use?	Yes	https://github.com/TripleS-org/PackTravel_G29
If you provide more extensive documentation, does this provide clear, step-by-step instructions on how to deploy and use your software?	Yes	https://github.com/TripleS-org/PackTravel_G29
Do you provide a comprehensive guide to all your software's commands, functions and options?	Yes	https://github.com/TripleS-org/PackTravel_G29
Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?	Yes	https://github.com/TripleS-org/PackTravel_G29
If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?	Yes	https://github.com/TripleS-org/PackTravel_G29
Do you store your documentation under revision control with your source code?	Yes	https://github.com/TripleS-org/PackTravel_G29
Do you publish your release history e.g. release data, version numbers, key	No	

features of each release etc. on your web site or in your documentation?		
Does your software describe how a user can get help with using your software?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does your website and documentation describe what support, if any, you provide to users and developers?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does your project have an e-mail address or forum that is solely for supporting users?	No	
Are e-mails to your support e-mail address received by more than one person?	No	
Does your project have a ticketing system to manage bug reports and feature requests?	Yes	https://github.com/TripleS-org/PackTravel_G29
Is your project's ticketing system publicly visible to your users, so they can view bug reports and feature requests?	Yes	https://github.com/TripleS-org/PackTravel_G29
Is your software's architecture and design modular?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does your software use an accepted coding standard or convention?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does your software allow data to be imported and exported using open data formats?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does your software allow communications using open communications protocols?	Yes	https://github.com/TripleS-org/PackTravel_G29
Is your software cross-platform compatible?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does your software adhere to appropriate accessibility conventions or standards?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does your documentation adhere to appropriate accessibility conventions or standards?	Yes	https://github.com/TripleS-org/PackTravel_G29

Does your software adhere to appropriate accessibility conventions or standards?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does your documentation adhere to appropriate accessibility conventions or standards?	Yes	https://github.com/TripleS-org/PackTravel_G29
Is your source code stored in a repository under revision control?	Yes	https://github.com/TripleS-org/PackTravel_G29
Is each source code release a snapshot of the repository?	Yes	https://github.com/TripleS-org/PackTravel_G29
Are releases tagged in the repository?	Yes	https://github.com/TripleS-org/PackTravel_G29
Is there a branch of the repository that is always stable? (i.e. tests always pass, code always builds successfully)	Yes	https://github.com/TripleS-org/PackTravel_G29
Do you back-up your repository?	Yes	https://github.com/TripleS-org/PackTravel_G29
Do you provide publicly-available instructions for building your software from the source code?	Yes	https://github.com/TripleS-org/PackTravel_G29
Can you build, or package, your software using an automated tool?	Yes	https://github.com/TripleS-org/PackTravel_G29
Do you provide publicly-available instructions for deploying your software?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does your documentation list all third-party dependencies?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does your documentation list the version number for all third-party dependencies?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does your software list the web address, and licences for all third-party dependencies and say whether the dependencies are mandatory or optional?	Yes	https://github.com/TripleS-org/PackTravel_G29
Can you download dependencies using a dependency management tool or package manager?	Yes	https://github.com/TripleS-org/PackTravel_G29

Do you have tests that can be run after your software has been built or deployed to show whether the build or deployment has been successful?	Yes	https://github.com/TripleS-org/PackTravel_G29
Do you have an automated test suite for your software?	Yes	https://github.com/TripleS-org/PackTravel_G29
Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?	Yes	https://github.com/TripleS-org/PackTravel_G29
Do you use continuous integration, automatically running tests whenever changes are made to your source code?	No	
Are your test results publicly visible?	Yes	https://github.com/TripleS-org/PackTravel_G29
Are all manually-run tests documented?	Not Applicable	
Does your project have resources (e.g. blog, Twitter, RSS feed, Facebook page, wiki, mailing list) that are regularly updated with information about your software?	No	
Does your website state how many projects and users are associated with your project?	No	
Do you provide success stories on your website?	No	
Do you list your important partners and collaborators on your website?	No	
Do you list your project's publications on your website or link to a resource where these are available?	No	
Do you list third-party publications that refer to your software on your website or link to a resource where these are available?	No	
Can users subscribe to notifications to changes to your source code repository?	No	
If your software is developed as an open source project (and, not just a project	No	

developing open source software), do you have a governance model?		
Do you accept contributions (e.g. bug fixes, enhancements, documentation updates, tutorials) from people who are not part of your project?	Not applicable	
Do you have a contributions policy?	Yes	https://github.com/TripleS-org/PackTravel_G29
Is your contributions' policy publicly available?	Yes	https://github.com/TripleS-org/PackTravel_G29
Do contributors keep the copyright/IP of their contributions?	Not Applicable	
Does your website and documentation clearly state the copyright owners of your software and documentation?	No	
Does each of your source code files include a copyright statement?	No	
Does your website and documentation clearly state the licence of your software?	Yes	
Is your software released under an open source licence?	Yes	https://github.com/TripleS-org/PackTravel_G29
Is your software released under an OSI-approved open-source licence?	Yes	https://github.com/TripleS-org/PackTravel_G29
Does each of your source code files include a licence header?	No	
Do you have a recommended citation for your software?	No	
Does your website or documentation include a project roadmap (a list of project and development milestones for the next 3, 6 and 12 months)?	No	
Does your website or documentation describe how your project is funded, and the period over which funding is	No	

guaranteed?		
Do you make timely announcements of the deprecation of components, APIs, etc.?	No	