IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

VOLUME : 5 ISSUE : 12 Print / Issue Publication Date: 13-Jul-2021

ISSN : 2455-2143

DOI : 10.33564/IJEAST.2021.v05i12.045

Indexed In

Google SCHOLAR OPEN ACCESS doi Crossref

WWW.IJEAST.COM

editor@ijeast.com

International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 12, ISSN No. 2455-2143, Pages 268-274
Published Online April 2021 in IJEAST (http://www.ijeast.com)

POOL: A PEER-TO-PEER RIDE SHARING APP

Ajinkya Ghorpade
Dept. of Computer Engineering,
NBN Sinhgad School of Engineering,
Ambegaon Pune 411041, India

Apurva Joshi
Dept. of Computer Engineering,
NBN Sinhgad School of Engineering,
Ambegaon Pune 411041, India

Sumit Mali
Dept. of Computer Engineering,
NBN Sinhgad School of Engineering,
Ambegaon Pune 411041, India

Piyush Agrawal
Dept. of Computer Engineering,
NBN Sinhgad School of Engineering,
Ambegaon Pune 411041, India

Harsh Agrawal
Dept. of Computer Engineering,
NBN Sinhgad School of Engineering,
Ambegaon Pune 411041, India

***Abstract*-- Ridesharing is becoming the most chosen transportation option over private transportation in big cities. Ridesharing is also very quick, affordable and safe. Ridesharing also helps lot of students who travel to college from very long distances and sometimes public transportation is also very limited. This helps them save both time and money and also helps them earn some money by listing their rides. So the paper presents the blueprint and application of a peer to peer or a dynamic pooling app which lets the users list themselves as drivers or riders and which allows them to find each other a ride within a specific area in particular bandwidth of time. The applications have some special features like fare recommendation according to distance, time, mileage and ride recommendation based on type of ride and the history of the rider.**

Keywords: **Peer-to-Peer, Pooling, Ridesharing, Price Recommendation, Pool.**

I. INTRODUCTION

(Liu, K. et *al*, 2019). Increase in population in metropolitan areas have increased the number of vehicles on the road resulting in an increase in environmental pollution. Nowadays ride sharing is becoming very popular almost everywhere. Ride sharing not only reduces the environmental pollution but also saves fossil fuels, time and money. It also benefits students in many ways. Students go to colleges from different remote locations and face difficulties in travelling due to lack of transportation facilities. So ride sharing helps them to travel to their desired location very easily. The ride sharing lets the users list their rides and also allows users to find a ride within a specific area. The features of the system are as follows:

Pooling: (Hasan et *al*, 2016). It is the backbone of the application and allows the users to add a new and also search for the available rides.

Select source and destination: The user can search through the list of locations around Pune city easily from the drop down suggestion box.

Peer to Peer communication: Passenger and driver will be able to communicate and finalize details like exact pickup spots and time.

Recommending fare: Fare recommendation is carried out according to distance, time and congestion at traffic hours according to which the user is travelling.

Recommending ride: Ride can be recommended to users according to the driver's history.

II. MATHEMATICAL MODEL

2.	Mapping Functions f(x)->y	X (Input)	Y (Output)
	F1(d) → F d -User data (username/password) (d ∈ D) F -Home page	d	F
	F2(F)→F'/R F'- Data record created	F	F'/R
	F3(F')→ dbs dbs - data in database	F'	dbs
	F4(dbs,R) → m m ∈ dbs(list of matching)	dbs,R	m
	F5(m,Ip3)→F'' F'' - ML model processing fare recommendation	m,Ip3	F''
	F6(F'')→Su Su- Final available list	F''	Su

Table 1 Mathematical Model

Mathematical model of our project describes complete working of the Rideshare app. Here f(x) denotes the function of x, where we have used f1, f2, f3, f4, f5 as functions with different inputs resulting in required outputs.

At the start of the app the user will be asked to login, this action is our function 1 'f1' with input 'd' as username and password and 'F' as output that is the Home page of the app. This can be denoted as F1(d)→F.

'F2' is the function in which the user creates a listing for the rideshare. This function takes in username and password (output of the previous function f1) as input and displays the listing as output. This can be viewed as F2(F)→F'/R where F' is the data listing.

International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 12, ISSN No. 2455-2143, Pages 242-246
Published Online April 2021 in IJEAST (http://www.ijeast.com)

After the user creates the listing, the ride schedule is then stored in the database. The function 'F3' takes F' as input and gives 'dbs' as output, denoted by F3(F') →dbs.

Now the rider searches his/her destination and the function 'F4' looks for the required details in the database resulting in a match found or not. This can be described as F4(dbs)→m where m is the result of a match found or not.

The next function 'F5' will be fare recommendation which will be done over the constraints like place, time, traffic congestion at that specific time. All the processing will happen using a machine learning model. This process is denoted by F5(m)→F''.

The final part of our app is to find the optimal listing. The data listed by the driver will be compared to data requirements of the rider. This is the function 'F6' denoted by F6(F'') → Su

III. SYSTEM ARCHITECTURE

System S is defined as collection of following set:
S = {Ip, Op, Ss, Su, Fi,A}

Mapping Functions f(x)	X	Y
F1(Ip1) → Op1	Ip1	Op1
F2(Ip2) → Op2	Ip2	Op2
F3(Op2) → Op2	Op2	Op2
F4(Ip2) → Op2	Ip2	Op2
F5(Ip3) → Op3	Ip3	Op3
F6(Op3)→Op3	Op3	Op3

Table 2 Mapping Functions

Objects:

1) Input1: Ip1 = {Username, Password}
2) Input2: Ip2= {Enter driver data (date, time, price)/Enter rider data(date, time)}
3) Input3: Ip3= {Enter destination}

1) Output1: Op1 = {Homepage (driver/rider)}
2) Output2: Op2 = {Data record created}
3) Output3: Op3 = {ML model processing for price recommendation+ List of available drivers}

Functional Dependency Graph:

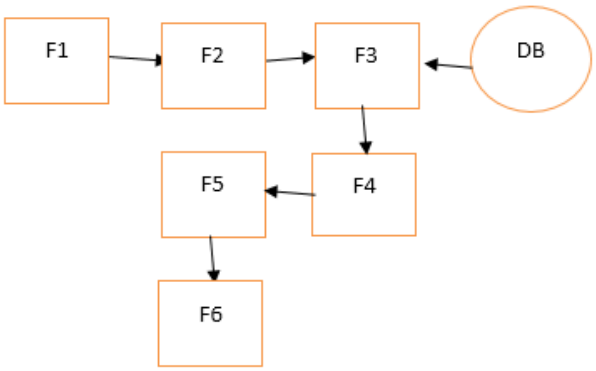

Figure 1 Functional Dependency Graph

(1) F1 = Login
(2) F2 = Enter driver data/rider data
(3) F3 = Data listing
(4) F4 = Time & location matching
(5) F5 = ML model processing
(6) F6 = Final available list

Below are the screenshots and explanations for our android based app.

1. **Login Page**

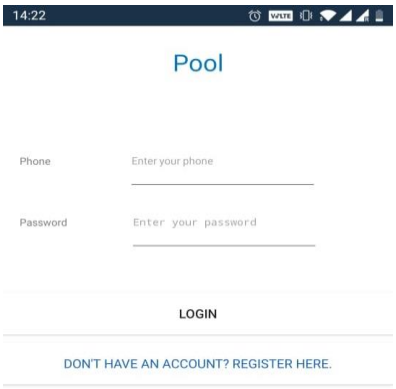

Figure 2 Login Page

This is the login page of the "POOL" app, where the user will enter their phone number as user id and password to login into their account. The GUI of this page is simple, two input fields for phone number and password, one login button to login and finally a link at the bottom to register as new users for creating a new account.

International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 12, ISSN No. 2455-2143, Pages 242-246
Published Online April 2021 in IJEAST (http://www.ijeast.com)

2.Registration page

Figure 3 Registration Page

For new users who don't have login credentials, the user is redirected to the registration page. For registration the user requires to enter a name, unique email id, unique phone number and password. Users have to select gender by radio buttons as it acts as the main feature of the app where further listings are filtered using gender.

For travelling from source to destination, the user has two options either to be a rider or a driver. Here driver means the person who will be taking a passenger while rider means the person who will ask for the lift. After successful login users can opt between rider and driver.

3. Driver Page

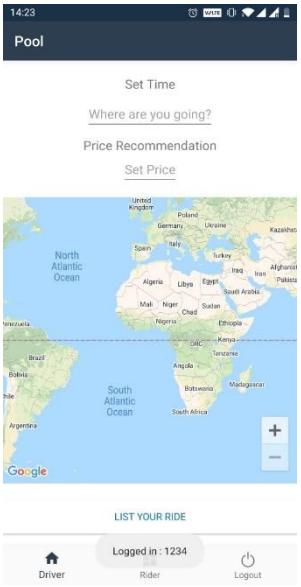

Figure 4 Drive Page

In the Driver page details required are like time of the ride or the location. This page contains Three input fields Set time for configuring the time of travel, selecting destination and confirming price of the journey. For visualization there is a map of the destination for better understanding of the place. In the bottom of the page List your ride button stored the data to the database and creates a listing for ride.

Figure 5 Example Screenshot

This screenshot describes the example where the driver puts in 3:30 pm time of travel, destination as Ambegaon Budruk and after the selection of destination distance of the travel with recommended price is displayed. The recommended price by the system is not the final price, the final price can only be set by the driver which can be haggled by the rider according to their understanding. Now at the final input field the driver will set the price he/she wants. In the map of the page the destination (Ambegaon Budruk) is set while the source being the same as NBN Sinhgad school of engineering.

4. Rider page

Figure 6 Rider Page

International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 12, ISSN No. 2455-2143, Pages 242-246
Published Online April 2021 in IJEAST (http://www.ijeast.com)

On the rider page, the user needs to select time of travel and destination. Our algorithm works in a way that it finds all the possible rides in a 30 minutes' time frame around the user's time. Users will be able to see a list of all available drives and can select from them according to time convenience.

IV. PRICE RECOMMENDATION SYSTEM

One of the important part of the app is price recommendation. That is when the driver enters the destination and time a certain fare is recommended according to the distance, time and traffic congestion. For this purpose, four different machine learning algorithms are compared and the best one is selected from: Random Forest, Gradient Boosting, Bagging and KNN.

1. **Random Forest:** (Jaiswal & Samikannu, 2017). Random forest is a supervised learning (A supervised machine learning algorithm is the algorithm that uses labeled input data to learn and predict when given new unlabeled data) algorithm for both classification and regression problems. Random forest creates multiple decision trees and combines them together to get a more accurate and stable prediction.

2. **Gradient boosting**: (Wen, et. *al*, 2018). Gradient boosting is a supervised machine learning algorithm for classification and regression problems, to produce a prediction model where it is an ensemble of weak prediction models, like decision trees.

3. **Bagging**: (Oza N.C, 2005). Bagging, also called Bootstrap aggregating, is a machine learning ensemble meta-algorithm for improving the stability and accuracy of machine learning algorithms used in statistical classification. It is also used to reduce variance and to avoid overfitting.

4. **K-Nearest Neighbor**: (Taunk, et. *al*, 2019). K-Nearest Neighbor is the supervised learning algorithm used for classification problems. K-NN algorithm finds the similar features between the data and available cases and puts the new data into the label that is most similar to the available labels.

These four algorithms are used to predict the fare of the ride according to the distance and time. Now this 'Time' is very important as according to the time of the day the traffic is different, for example at evening hours when the office timing ends the traffic is very high while at morning or afternoon we can see less traffic hence according to time the fare changes. To achieve this, we used four machine learning algorithms with a dataset containing congestion, location, distance and fare as features and trained the model. The model checks the time and predicts if there will be congestion or not and hence displays the recommended fare. As we have discussed earlier, time is a very important feature which will decide the rush hour or congestion in the traffic hence the fare will be decided accordingly.

V. SOFTWARE MODELLING

1. USE CASE DIAGRAM

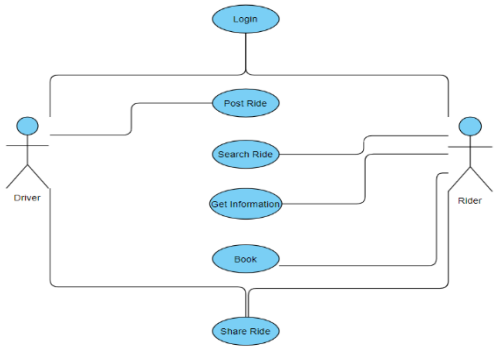

Figure 7 Use Case Diagram

Use case diagram depicts the explanation from system architecture and shows the relation between driver and rider. Here after successful login the user selects either to be a rider or a driver. If driver is selected then the driver will post the ride while if rider is selected then the rider will search the ride, get information and finally book the ride. Now if everything matches they will share the ride.

2. ACTIVITY DIAGRAM

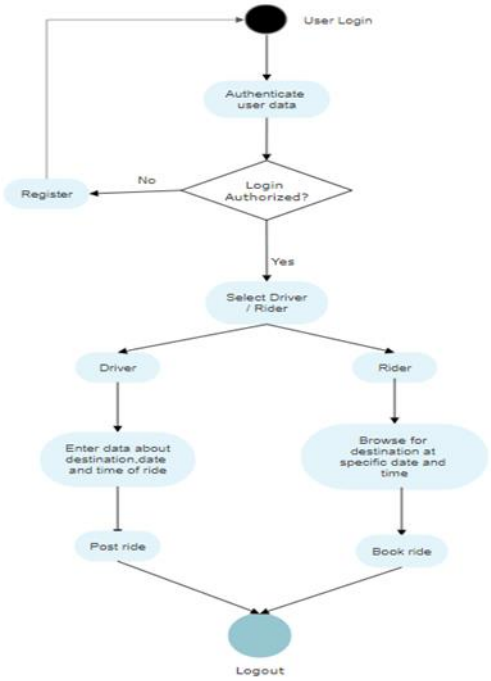

Figure 8 Activity Diagram

International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 12, ISSN No. 2455-2143, Pages 242-246
Published Online April 2021 in IJEAST (http://www.ijeast.com)

Activity diagrams represent workflows in an graphical way. They can be used to describe business workflow or the operational workflow of any component in a system. Sometimes activity diagrams are used as an alternative to State machine diagrams. Check out this wiki article to learn about symbols and usage of activity diagrams.

3. COMPONENT DIAGRAM:

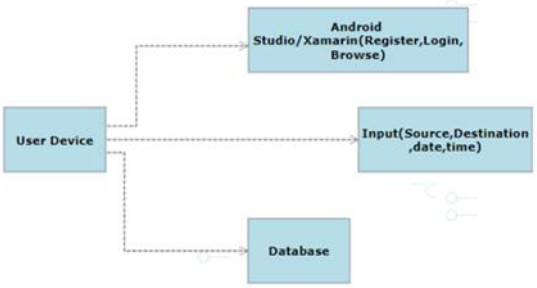

Figure 9 Component Diagram

A component diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems that has many components. Components communicate with each other using interfaces. The interfaces are linked using connectors. Below images shows a component diagram

4. DEPLOYMENT DIAGRAM:

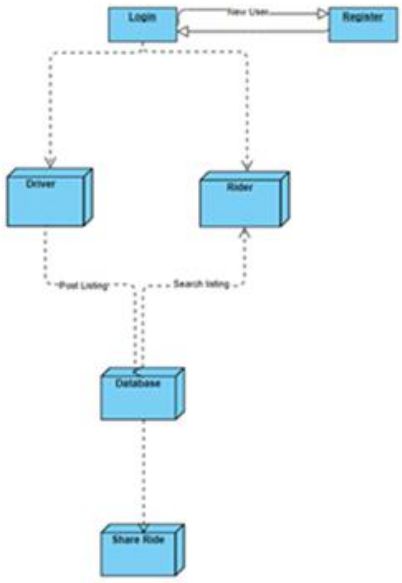

Figure 10 Deployment Diagram

A deployment diagrams shows the hardware of your system and the software in those hardware. Deployment diagrams are useful when your software solution is deployed across multiple machines with each having a unique configuration. Below is an example deployment diagram. UML Class Diagram with Relationships

5. CLASS DIAGRAMS:

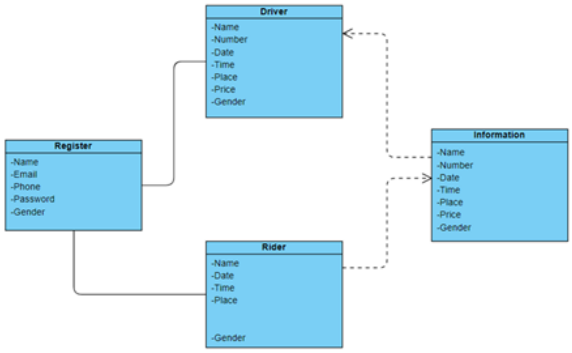

Figure 11 Class Diagram

The purpose of a class diagram is to depict the classes within a model. In an object oriented application, classes have attributes (member variables), operations (member functions) and relationships with other classes. The UML class diagram can depict all these things quite easily. The fundamental element of the class diagram is an icon the represents a class. This icon is shown in the figure above.

VI. SOFTWARE TESTING

(Gaur, et. *al*, 2016). Software program testing is the procedure of comparing a software program object to come across variations among given input and predicted output. Also to evaluate the characteristics of a software program item. Software testing is a method that ought to be accomplished at some stage in the development process. In different phrases software program testing is a verification and validation method.

Verification: Verification is the process to make certain the product satisfies the conditions imposed on the start of the development phase. In different phrases, to make sure the product behaves the manner we want it to.

Validation: Validation is the technique to make certain the product satisfies the required requirements on the give up of the development phase. In different phrases, to make sure the product is built as consistent with client necessities

Basics of software testing: There are two basics of software testing: black-box testing and white-box testing.

International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 12, ISSN No. 2455-2143, Pages 242-246
Published Online April 2021 in IJEAST (http://www.ijeast.com)

- **Black-box Testing**: Black box testing only focuses on the output generated and ignores the internal structure of the system.

- **White-box Testing**: White Box testing or the structural testing focuses on the internal structure of the system and is always used for validation purposes.

Test Case ID	1
Test Case Description	Run Xamarin code on virtual machine
Steps	1. Debug and Compile the code. 2. Run the code.
Test Case Result	Application should be successful installed.
Action Result	Application successfully installed and started.
Status	Pass

Test Case ID	2
Test Case Description	Applications home page should get displayed on start-up with option of login and registration.
Steps	1. Open application 2. Register 3.Login
Test Case Result	Application should be successfully started and Register Form should be accessible.
Action Result	Application accepted registration data and able to login successfully.
Status	Pass

Test Case ID	3
Test Case Description	Selection between driver and rider
Steps	1. Login 2. Select driver or rider
Test Case Result	Display the respective page for driver and rider.
Action Result	Both pages are displayed after each selection test.
Status	Pass

Test Case ID	4
Test Case Description	Processing of data for driver page
Steps	1. Select as driver 2. Enter the required data like time, place, and price. 3. Submit the data 4. Get recommended price
Test Case Result	Data is stored in database and listing is created. Recommended price is generated.
Action Result	Data successfully stored and listed
Status	Pass

Test Case ID	5
Test Case Description	Processing of data for rider page
Steps	1. Select as rider 2. Enter the required data like time and place. 3. Submit the data
Test Case Result	Data is compared with the available rider data in database.
Action Result	Data compared and match was found for the ride.
Status	Pass

Table 3 Test Cases Table

VII. FEASIBILITY STUDY

1. Technical feasibility

(Standing et *al*, 2018). Software- We have used java which is highly scalable and widely available. It runs on a variety of devices like smartphones and smart watches etc. So that our app can be deployed on them.

Technical skills & team members -Skill required are App development and Database management with a team of four.

Hardware - Pc with min spec core i3, 8gb ram, 256 GB HDD and Xamarin / Android Studio (Application development platform) .

2. Operational feasibility

Usability -Handling text data does not require big data space .The app will run on our private server that can handle text related data very easily.

Efficiency - App size is less than 20 Mb and will run on Android 2.3(Gingerbread) and above.
Maintenance - Very low maintenance is required with few updates accordingly.

Automatic gender pairing: The app automatically displays information about drivers according to gender which provides a safety and comfortable space to share rides.

3. Economic feasibility

Design and Development cost - As we are using open source and freeware softwares, the design and development cost is negligible. This gives us an advantage of spending less on licensed software which lowers our cost of development.

Operational cost - It includes the private server maintenance cost which is around 8000 every year.

Deployment cost - The cost of deployment is from 2000 to 5000.

4. Legal feasibility

Data protection - We only keep registration data and do not record our day to day rides as our operation model is based on temporary transactions.

VII. CONCLUSION

In this paper, we have explained about the use cases ,working and system architecture of our peer to peer ridesharing application.This project finds an alternative to public transportation ,replacing it by peer to peer ride sharing. Our application tackles various issues involved in public transportation by using shortest path algorithms , p2p communication, optimal routing and machine learning based price recommendation. Similarly by addition of functions like payment gateway, realtime notification service this application can be revised. There can be many more functions like this which we hope to add in POOL.

VIII. REFERENCES

[1] Liu, K., Zhang, J., & Yang, Q. (2019). Bus Pooling: A Large-Scale Bus Ridesharing Service. IEEE Access, 7, 74248–74262. doi:10.1109/access.2019.2920756

[2] Hasan, R., Bhatti, A. H., Hayat, M. S., Gebreyohannes, H. M., Ali, S. I., & Syed, A. J. (2016). Smart peer car pooling system. 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC). doi:10.1109/icbdsc.2016.7460384

[3] Dejan Dimitrijevic, Vladimir Dimitrieski, N. Nedic (2013), Real-time carpooling and ride-sharing: Position paper on design concepts, distribution and cloud computing strategies Computer Science 2013 Federated Conference on Computer Science and Information Systems.

[4] Binu, P. K., & Viswaraj, V. S. (2016). Android based application for efficient carpooling with user tracking facility. 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC). doi:10.1109/iccic.2016.7919536

[5] Craig Standing, Susan Standing & Sharon Biermann (2018): The implications
of the sharing economy for transport, Transport Reviews, DOI: 10.1080/01441647.2018.1450307

[6] Gedam, C., Sahare, M., Sachdeo, R., & Kulkarni, N. (2020). Smart Transportation Based Car Pooling System. E3S Web of Conferences, 170, 03004. doi:10.1051/e3sconf/202017003004

[7] Jaiswal, J. K., & Samikannu, R. (2017). Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression. 2017 World Congress on Computing and Communication Technologies (WCCCT). doi:10.1109/wccct.2016.25

[8] Wen, Z., He, B., Kotagiri, R., Lu, S., & Shi, J. (2018). Efficient Gradient Boosted Decision Tree Training on GPUs. 2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS). doi:10.1109/ipdps.2018.00033

[9] Oza, N. C. (n.d.). Online Bagging and Boosting. 2005 IEEE International Conference on Systems, Man and Cybernetics. doi:10.1109/icsmc.2005.1571498

[10] Taunk, K., De, S., Verma, S., & Swetapadma, A. (2019). A Brief Review of Nearest Neighbor Algorithm for Learning and Classification. 2019 International Conference on Intelligent Computing and Control Systems (ICCS). doi:10.1109/iccs45141.2019.9065747

[11] Gaur, J., Goyal, A., Choudhury, T., & Sabitha, S. (2016). A walk through of software testing techniques. 2016 International Conference System Modeling & Advancement in Research Trends (SMART). doi:10.1109/sysmart.2016.7894499

[12] A Guide to the Project Management Body of Knowledge (PMBOK® Guide) Third Edition, Project Management Institute, 2004

IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

ABOUT IJEAST

International Journal of Engineering Applied Science and Technology (IJEAST) is a peer-reviewed, open access journal that publishes high-quality research papers in the field of Engineering, Applied Science and Technology.

IJEAST aims to provide a platform for researchers, academicians, and professionals to share their innovative ideas, research findings, and practical experiences with the global scientific community.

FOCUS AREAS

- Engineering
- Applied Science
- Technology
- Innovation & Development
- Interdisciplinary Studies

PEER REVIEWED
All submissions are rigorously peer reviewed to ensure quality.

OPEN ACCESS
Free and unrestricted access to research for all.

GLOBAL REACH
Connecting researchers and professionals worldwide.

TIMELY PUBLICATION
We ensure a swift and efficient publication process.

For more information, visit our website
www.ijeast.com

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

editor@ijeast.com
www.ijeast.com
India

2455-2143