

B.TECH. PROJECT ON

FTV – (Feel the Virtual)

Submitted By

Shivam Sharma (2019UCO1573)

Rijul Garg (2019UCO1570)

Vishal Verma (2019UCO1563)

Shrey Rastogi (2019UCO1562)

Under the Guidance of

Sh. Rajeev Kumar



Division of Computer Science and Engineering

NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

NEW DELHI – 110078

Sep - Dec 2022

Project Description

With coming into the new era Online Shopping is coming out to be one of the biggest revolutions in the today's world and according to the survey about 63% of the global population have internet access in 2022 but when compared to the population of user who tries to buy the stuffs online this number decreases.

The main reason for this decrease is due to fact that people opt for visiting the shop offline is because the demand to view the product by themselves, to have the feel of the texture of the object.

People do hesitate while shopping online thinking what if this shirt doesn't fit on them well what if the sofa or almirah will look good in their drawing space. People always demands to interact with the products.

In order to solve this problem of Online Customer-Buyer relationship, the new technology known as Augmented Reality is in the market for quite a few years can be used. In a survey conducted by Google, around 66% of the users showed interest in Augmented Reality for shopping.

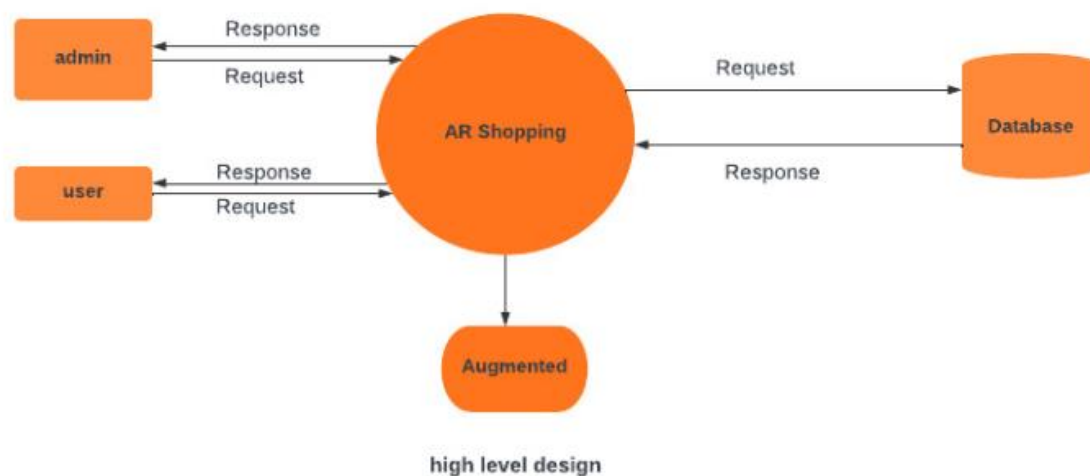
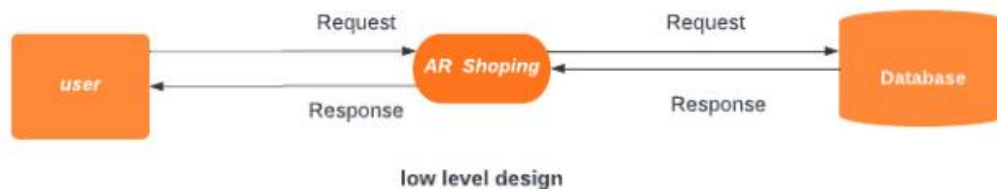
Many different applications such as snapchat, Instagram uses the AR technology of superimposing different objects pr filters on the face. Many other brands such as Nike, Lenskart allows the user to virtually try their product on themselves. But the problem with these applications is that each application based on one type of product like Nike *for shoes* or Lenskart *for glasses* etc.

Our project is an e-commerce application harnessing the power of AR technology. Project includes the Web application (*for the admin side in order to manage the products*), Android application (*for the user side to view the product with voice search feature allowing them to have easy purchases*) and with the AR feature (*for the user to view the product online and have the virtual experience*).

The main application of our product is for the E-commerce side. As the Covid-19 pandemic has struck and changed the life of the humans drastically. Now with the new global scenarios use of online applications specially shopping apps like Amazon, Myntra, Lenskart and many others had increased a lot.

People tends to use these apps and buy the groceries, clothes, appliances at their door steps only. But while shopping online there is always a hesitation in the user whether the product will fit in their house, whether the color looks good on them or not. In order to bridge this gap between the customer and buyer relationship our application **FTV** (*Feel The Virtual*) is the best application to have a lively and immersive shopping experience with 3-D visualization, virtual try-ons and product demos.

This will increase the engagement of user on the product page and leads to the customer satisfaction and also reduces the return rates which one of the major problems of online shopping nowadays that people are not happy with the product they buy, which can be because the size of the product or the color of the product or other reasons. And this feature of the app to have the lively experience of the product leads to the increase in the conversion rates.



Plan of Action/ Phases of Project

Our project is mainly divided into 4 phases. Each of the phases are connected to each other and requires the complete integrity.

First and the Second Phase – Our project starts with **Building the AR system**. Our AR system will be build using Unity 3d Software and Vuforia. Vuforia is the AR software development kit for the android devices used to build AR application. Software is based on the computer vision technology and can be used to project the augmented/virtual object (*here the object can be any 2d or 3d object or it can be any video*) on the target object, here the target object can be any 2-d photo or the empty space. The image recognition system of the Vuforia will detect the target image using the CV technology by finding gradients of the target image and when it detects the image then it will superimpose the augmented image or video on our target image.

Third Phase – Third Phase of our project will be started with **Building the mobile and web app** Our Admin page will allow the admin to manage the product details, user details in the database and keep the track of each transaction of each purchase made by the users. One thing to notice that here there can be more than one admin. This can be build using the normal Web stack like HTML, CSS, JavaScript (for the frontend) and Node.js and express.js and MongoDB as database (for backend).

Our Client-Side Application is the mobile application which will allow the user to view the products. This Mobile Application will be build using Unity 3d.

Final Phase – At last comes the **Integration of the project** The last and the final phase of our project is toh integrate our AR application with the web & mobile application that we built and make it all one integrated system.

Expected Deliverables of our Project

- (1) Mobile App** – This Mobile Application will be build using 3D Unity and it basically shows the list of items that are available at the buyer/ store. User here can scroll different products and can try them using the AR feature available on the app.
- (2) Admin Application** – This Admin Application will be used to manage the products on the E-commerce website. It basically allows the admin to add, remove and update the product count on the server.

Outcome of Discussion with Guide

After the discussion with our guide, we had the following outcomes:

- The major emphasis was on more research on the AR aspect of the project as we are currently not very well versed with this technology.
- The admin side should have the functionality of adding new objects, removing new objects.
- There should be proper synchronization between the web application and android application via the database.

Software Using –

- Unity 3D
- Vuforia
- ARCore SDK
- HTML/CSS, JS (Frontend)
- MongoDB (Backend)

Responsibility of Each Student

The complete project work is distributed among the four students as follows:

Shivam Sharma - will work on the AR model. As we are new to the AR world, so we tried first go for the research work and try to study about the new technology and all the key elements that are required for the smooth functionality of our model. After all this we will make an AR model using unity and Vuforia. That will be integrated will are mobile and web app at the backend side.

Shrey Rastogi - Will work on the mobile app. That is our main user interface where our user can come and select a product which he wants to test.

Vishal Verma - Will work on the admin side site where the admin would be able to add the new product that he wants the user to view. The admin would be able to add, edit, delete all the products etc.

Rijul Garg - will work on the database side where all the product details, user details, dealer details etc. are stored in a MongoDB server and the server will we all the necessary DBMS functionality like CRUD operation and ACID properties.

Progress Report of the Team (As Plan of Action)

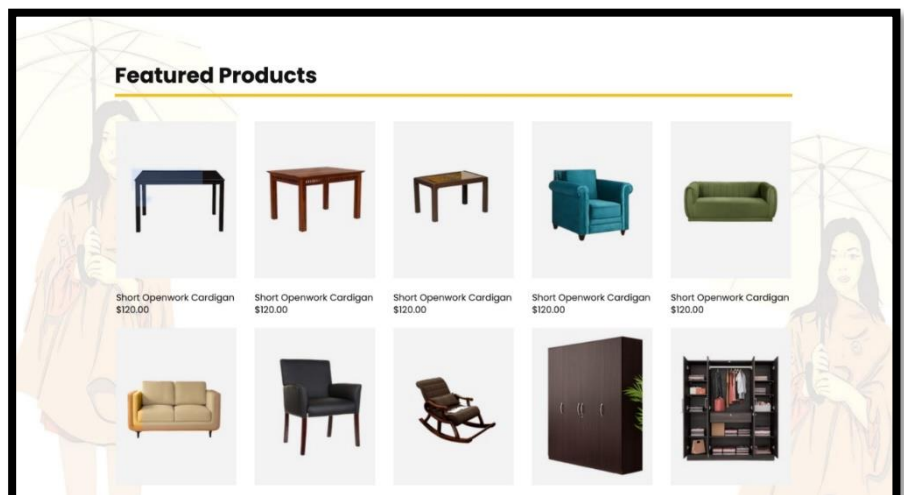
Our project has the main emphasis is on Building the E-commerce application which can emphasizes on improving the user experience by adding the AR functionality in it. So, our project is divided into two segments - one is Building the e-commerce application for the user experience and the second one is Adding the core functionality of AR into it. So, we divided our team also into two units, where one unit will work on the functionality of AR because of the research work it included in it and the second team will build the ecommerce application for the user and the admin control system where they can update the items in the warehouses according to the retailer's stock value.

For Building the AR functionality, our team first tried to built it using Java and android studio but the problem with it that the application can't be used in the IOS devices. So, after completing the research on the AR functionality and there use cases we agreed to build the app using the Unity – 3D and Vuforia which comes with the lots of free objects/items to check for the functionality of the AR application. So, after completing our research we tried to build the small demo prototype to check the functionality of the app, so for that we use the small object, a plant and tried to check whether the application is successfully detecting the target image and will it able to superimpose the plant on that target image or not. For now, the object is successfully superimposed on the real-world place but we still needed to improve the image detecting algorithm of the application because the app still not be able to work properly if we move the camera little bit.



For the second and the most important part of our project is to build the e-commerce application for user as well as the admin side. So, for now we are concentrating on building just web interface where user can interact with the products which are pushed by the retailer on the application.

So, For Building the web application we are using the HTML/CSS and JavaScript for the frontend and at the backend we are using the Node.js and for the Database since we are working in the production environment, we are hereby using the local Database of MongoDB but in the future once we deploy our application over the internet then we will extend it to MongoDB or the Firebase Database. Till now, we had built the main page of our application where user can see the product details and their retailers, and an Admin page where the admin can add or remove the retailers' details as well as product.



Contribution of all members

Vishal Verma

As the role divided among the students, I've been into the web dev team. I'm responsible for building the web interface for the admin side, where admin can easily add or remove the stock details as well as retailer's details associated with the selling their products. For now, in collaboration with another team member Rijul Garg we had worked on the web interface for the user too. Here user can simply view the product details available on the store and can add the products in their Wishlist.

We've built the application using HTML/CSS, Bootstrap and JavaScript for the frontend in order to make the website more colorful and interactive and used Node.js and Express.js at the backend of the application. And till now, we successfully connected our application with Database so admin can easily can make add or remove the stock details. Database we are currently using the local Database of our System using MongoDB.

Initially admin has to login to the system using their credentials and after login, the interface shows them to option one to watch the stock details and their retailers' details. And second tab will lead them to the form, in the form it basically asks them to add the product name, one picture of the product and their details and the retailer's details who is selling the product. Once the product is added to the database it will render them to the page showing details of all the product which are now available in the warehouse.

For the client side, we just show details of all the product details and the order details by the user.

Rijul Garg

As the role divided among the students, I've been into the web dev team. I'm responsible for building the web interface for the client side, where user can easily view the product available in the warehouse. For now, in collaboration with another team member Vishal Verma we had worked on the web interface for the admin too. Here admin can add the product details and the retailer's detail into the database. We've built the application using HTML/CSS, Bootstrap and JavaScript for the frontend in order to make the website more colorful and interactive and used Node.js and Express.js at the backend of the application. And till now, we successfully connected our application with Database so admin can easily can make add or remove the stock details. Database we are currently using the local Database of us System using MongoDB.

For the client side, we just show details of all the product details and the order details by the user. All the product info and their prices along with some product description we added on the page. User can also get more info regarding the product details and the retailer's detail once clicks the product image.

Initially admin has to login to the system using their credentials and after login, the interface show them to option one to watch the stock details and their retailers details. And second tab will lead them to the form, in the form it basically asks them to add the product name, one picture of the product and their details and the retailer's details who are selling the product. Once the product is added to the database it will render them to the page showing details of all the product which are now available in the warehouse.

Shivam Sharma

I worked together with Shrey to build the application and knowing the AR technology. The heart of the AR technology lies in understanding how the scene is interpreted and information is extracted from it.

For that we need some understanding of **image processing algorithms** that work behind the scenes. One of the most fundamental concepts involves anchors. These rely on key points and their descriptors, detected in the recording of the real world.

What is a feature point? It's a distinctive location in images – for example corners, blobs or T-junctions are usually good candidates. The most important property of a good feature point is **reliability**. The algorithm must be able to find the same physical interest point under different viewing conditions. A lot can change while you're using an AR app:

- camera angle / perspective
- rotation
- scale
- lightning
- blur from motion or focusing
- general image noise

Initial algorithms for finding feature points like SIFT, SURF is too slow for real-time use on mobile devices. Although it is not known which underlying algorithm is being used in today's AR SDKs but a possible candidate is “**BRISK**” (“*Binary Robust Invariant Scalable Keypoints*”) algorithm by **Leutenegger et al.** It's fast and efficient enough to serve as base of the overall “**SLAM**” approach of simultaneously locating the camera position and mapping the real world.

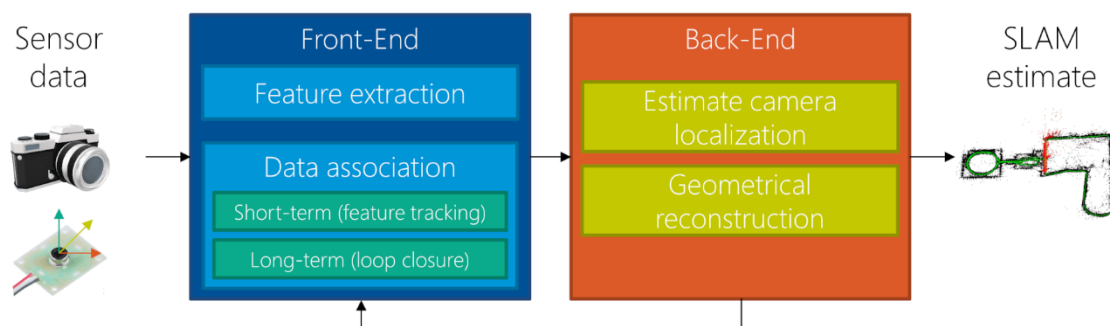
For Augmented Reality, the device has to know more than just its key point descriptors: its 3D position in the world. It calculates this through the spatial relationship between itself and multiple key points. This process is called “Simultaneous Localization and Mapping” – SLAM for short.

The algorithm has two aims:

- 1) **Build a map** of the environment
- 2) **Locate** the device within that environment

To make Augmented Reality work, the SLAM algorithm has to solve the following challenges:

- Unknown space.
- Uncontrolled camera. For current mobile phone-based AR, this is usually only a monocular camera.
- Real-time.
- Drift-free.



Basic overview of the SLAM algorithm

Current trends: - what's used in today's mobile AR? All fuse data of inertial sensors together with the camera feed.

- a) **Google:** In the documentation, Google describes that **ARCore** is using a process called concurrent odometry and mapping – which is essentially just another name of the broader term SLAM.
- b) **Apple:** several years ago, Apple acquired *Metaio* and *FlyBy*, who developed SLAM algorithms. Apple is using **Visual Inertial Odometry** / SLAM – so they also combine the camera with other phone sensors to improve precision.

Shrey Rastogi

I worked simultaneously with Shivam on the working of AR and building the mobile application. We built a basic initial mobile application in Unity 3D using C# as Unity provides platform support for both IOS and Android (*unlike Android Studio*). The Unity Editor is a popular and useful authoring platform to create cutting edge augmented reality experiences for handheld devices. Vuforia Engine is a software development kit (SDK) for creating Augmented Reality apps.

The app fixes a plant object (*in 3D*) on a target object according to the user's wish. What the app does can be summarized in the following steps:

- 1) The user clicks an '**instant target image**' by clicking the *camera icon* at the footer of app.
- 2) The image is then scanned for feature points or **key point descriptors** (*some examples of features points are marked in orange in the picture on right*). These features also need to be associated with landmarks – key points with a 3D position, also called map points.
- 3) Afterwards, the plant is then placed on the object having the most key points i.e., it can be recognized distinctly from other features in the image with ease.
- 4) As we change the viewing angle, our view of the plant also changes, the plant finds the best orientation in 3D space to put itself on the object as preserves that orientation even when the camera angle and direction of view change.



Output of Discussion with the Guide

After having the deep discussion with the Guide, we are presented with some of ideas to improve the functionality of our application. Some of them were –

- (1) Try to improve the Object Detection Algorithm of the AR application that was built. Since, Vuforia uses the SLAM algorithm for object detection, our guide has encouraged us to do a more research on the algorithm and how can we improve the object detection of the system since there is a bit issue when the object is moving.
- (2) Research on how to add the connectivity between the mobile application and the web application. Connect the product database i.e., web application with the AR application till now. So, that as soon the admin updates the stocks or remove the product from the database the user can see the difference on the mobile app.
- (3) Make the web interface more interactive and colorful and easy, so user knows why to use our particular application instead of using the other product. Try to add more useful functionality in the application.
- (4) Do the research work on how to add the functionality in the application such that when the user is offline, he is still able to see the product details as this functionality is missing in many modern e-commerce applications which we see nowadays like amazon and Flipkart.
- (5) Try working on the free online Databases available over the internet like MongoDB Atlas and Firebase database.

Mid-semester examination review by examiners

1. The performance of application on very distinctive type of objects (as compared) to surroundings is good but try to improve placing models on not-so-distinctive objects.
2. Research on how to add the connectivity between the mobile application and the web application. Connect the product database i.e., web application with the AR application till now.
3. How would you make the website lighter
4. Extend the functionality of app to multiple such use cases.

How these suggestions were incorporated into project –

1. To improve performance of object detection, we switched from using the Vuforia Engine to the ARfoundation framework that is provided via Unity's AR Foundation package.
2. We used Vuforia for our purpose which provide marker-based AR capabilities and relies on third party servers for AR capabilities; along with this marker-based AR technology it also had some implicit flaws
 - first of them was that when we were using camera feed alone to provide AR experience which was slightly inaccurate. There were some jitters in the object placed, it was not stable where it was placed.
 - We relied on image or marker being in the view to visualize object and due to this we were not able to see object from all directions as some perspective cause marker being out of site.
3. Because of these problems in marker-based AR we researched for alternative solutions available in the market and as a result we are using AR foundation provided by unity for the AR capabilities which uses platform-based technologies to support AR capabilities. One of the key benefits of this is that

it does not relies on third party server to provide AR capabilities. It uses platform dependent capabilities which are provided implicitly by the device itself. Such as ARcore provided by android ARkit provided by IOS. Along with this this technology supports or make use of marker less AR technology which solved both of our problems faced in Vuforia: -

- Object placed in the real world are very stable
- It makes use of other sensors data too to localize device in the real world and due to this we can visualize our object from all directions.

4. For the connectivity part between the mobile application and web application, we have used the Firebase Realtime database provided by Google.
5. Since the previous website we have designed was built using express and Mongo DB but the main problem was how to store the AR model on the server- side database as it was not possible the huge file on the Atlas Server so, we have decided to switch to the Firebase as it allows us to easily store the huge files as our requirement. Hence, we changed our Database Mongo Atlas to Firebase.
6. Since Firebase doesn't support the Express and Node.js at the backend, we decided to switch to React.js which is one of the easiest frontend frameworks of JavaScript.

TECHNICAL WORKING

The project can be broken down into five major sections:

1. User's side of the e-commerce application
2. Administrator's side of the e-commerce application
3. Android Application for the user
4. Creating 3D assets for the AR
5. Developing the web-based AR application

ANDROID APPLICATION

Talking about the app, we have developed a complete **pipeline** which does all the necessary task.

- **Fetch product list from firebase** real-time database and populate product in our app.
- Each product contains product details along with links to the image of the product and a link to the AR model.
- Fetch product image from the firebase storage capability.
- For the model we look for the model in **local cache** and if it is not found we download it from firebase storage capability and **cache it for next time to save some network bandwidth**.
- Key point to note here is that models are larger than other data about product that is why we have **optimized** this. We fetch the model only when the model is required or basically when user wish to test that product in the room.
- Then we preprocess the model to show in AR.

Note that any random model cannot be uploaded to the server. This model needs to full fill some criteria for which we have a **fixed format** - We can only upload the *AssetBundle* to the website which have to fulfil following requirements.

- It can use any 3D model scaled according to the real-world size. For reference 1 unit in unity is 1m in real world.

- It must contain model as **single mesh**. With single mesh renderer.
- Apply necessary textures and materials
- It must contain one **mesh collider**.
- It must be oriented correctly as orientations in unity and various 3d modelling tools are different.

Once the model fulfills all the requirements, we need to follow these steps to get final output *AssetBundle*:

- **Create prefab** out of the model
- **Assign a name** to the *AssetBundle* in the prefab window
- Then run the *AssetBundle creator script* using custom created menu option in Asset menu.
- Final *AssetBundles* can be found in ***AssetBundle folder*** created by the script

This *AssetBundle* can be uploaded on the server of database by the vendor. Note that all of the above-mentioned tasks could be easily done by the product manufacturer as all product goes through design process and during that models are created.

Vendor need to simply get the *AssetBundle* from the manufacturer and upload it. Another important aspect is the preprocessing before the model can be displayed in the AR view.

This is the common pipeline where *AssetBundle* is used to create *gameObject* and we add few components which are used in AR mode to interact with the model.

- We create **child mesh** slightly bigger than original product which is used to provide feedback to the user that whether user can interact with the object or not
- *ARSelectable* component is added which **looks for the inputs** and toggles this feedback mesh
- *ARRotation* Handler component is added to **control rotation** capabilities.

After adding all these components model is ready to be viewed in the real world but we need to configure AR capability to know exactly what kind of interactions are expected for the object which include whether object is any object which can be places on floor or could it be hanged on the wall etc.

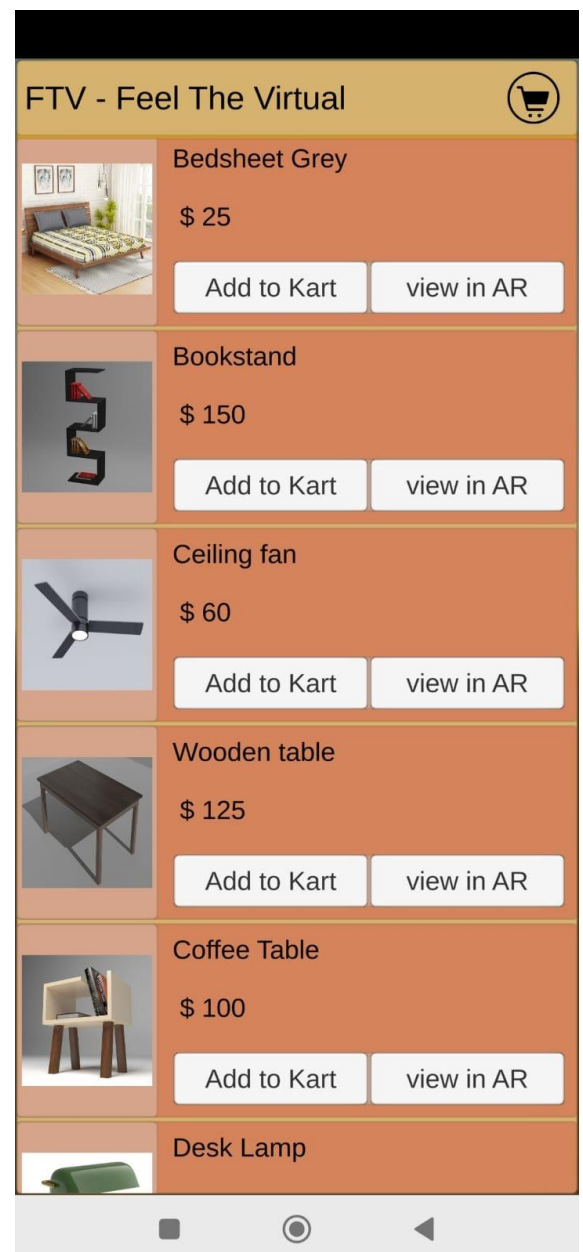
Working of AR

Components like *AR Session* and *AR Session Origin* are used to manage the task of **localization** of the device in real world. *AR Session* make sure that it knows its orientation, location and motion in real world based on various data sources such as camera and various sensors.

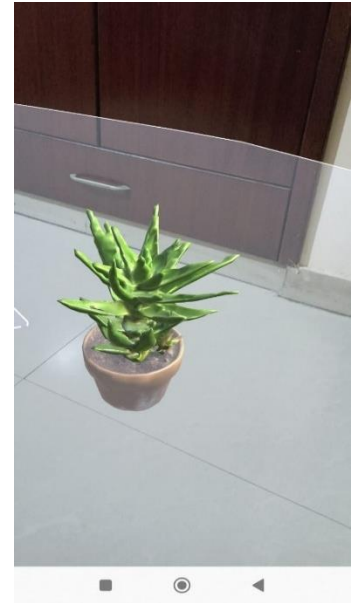
AR session origin make all the necessary calculation to **map real world** and the virtual world based on the device location and orientation.

A component called *AR Plane Manager* is used for maintaining all the heavy load of

- Looking for the **feature points** to get information where or at what height a **plane** could be created based on the real-world surfaces
- Other **sensor's data is used to localize** itself in real world and to track its location in real world. This data is used to know its orientation and to decide **plane's orientations.**
- Then planes are created by the **plane manager**. They are merged with other planes if they are found to be on same level and all other things are managed about the plane.



A component called *Object Placement Interactions* make sure that the product is placed on the right place in the world. This is done by shooting a **raycast** to get the location of the plane where the object could be placed and then based on the results derived from raycast are used to place the object.



THE ECOMMERCE SITE

The complete Web Application is build using the technologies HTML, CSS , JavaScript and React for the Frontend and using the Firebase for handling the backend. Initially the backend was built using the Node and Express.js and MongoDB as the database but as proceeding with the project we found that storing the AR model file in the local storage makes our app bulkier and takes time to load the data from the database and decrease the performance of the app. So, hence we decided to migrate our application working to the React in the frontend and opting for the cloud service of Firebase which decreases the load on the app and increasing the performance of whole application.

Why React and not the Node and Express.js?

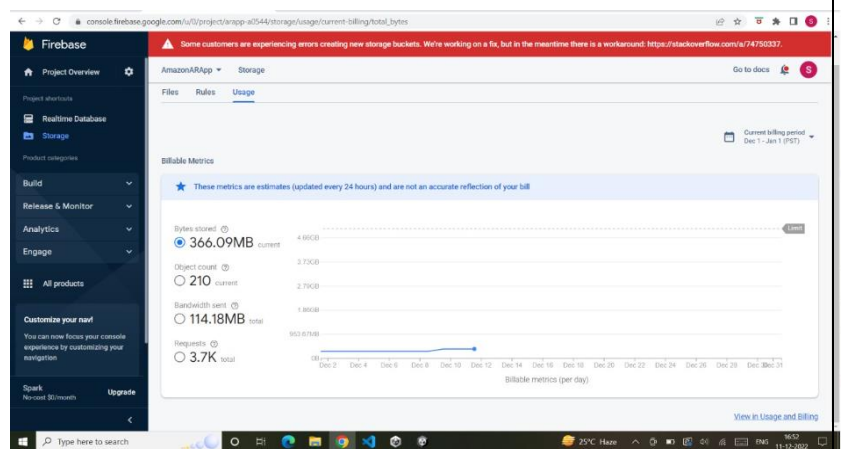
(a) Speed. The ReactJS framework allows the developers to use different components on the client as well on the server side without causing rippling changes throughout the codebase. Since loading the AR model everytime when we display the list of all the product is the heavy work hence using the ReactJS will improve the speed of the application.

(b) Can Easily be shift from web to mobile. Once a business scales, it is likely that it would need a mobile app. And React facilitates an easy shift from web to mobile, as the code written in React is compatible with React Native.

(c) Easy to update and Manage the whole application. apps that are built on ReactJS are flexible and can be scaled easily. Moreover, it also saves a lot of developers time, money, and effort in the long run.

Why Using Firebase instead of MongoDB and MongoDB Atlas?

(a) Different services like Real Time Database, Storage Service and the Authentication services is provided by the Firebase server and it supports to connect with different types of projects such as web, IOS and Unity.



Firebase Real Time Database

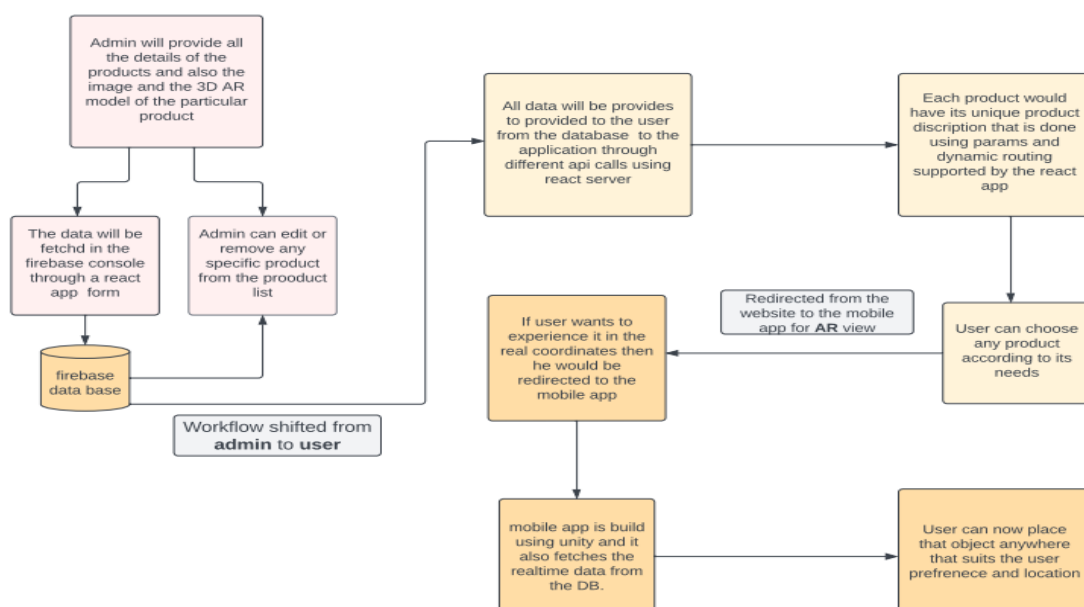
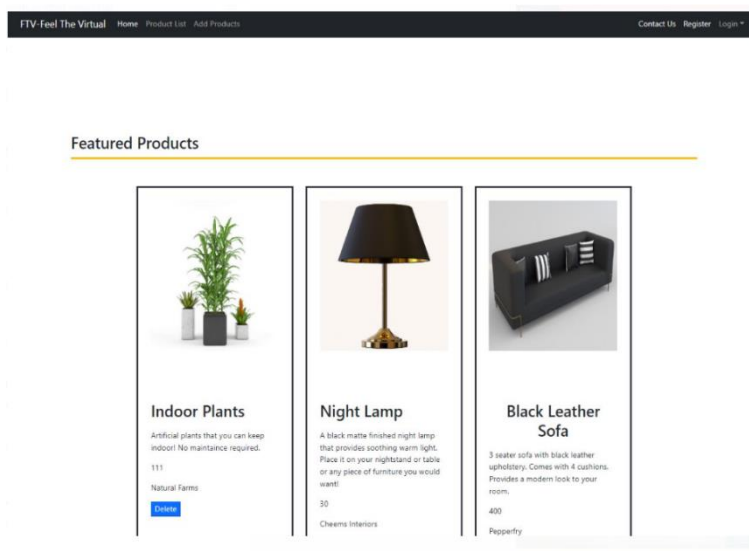


Diagram describing the overall workflow of the ecommerce site

ADMINISTRATOR'S SIDE OF ECOMMERCE PLATFORM

At the admin side we will have a proper authentication for the login purpose and the admin will be provided a key that they can use for the signup purpose. now the admin would be provided with the image and 3D model by the vendor who wants his product to be featured on the site. then the admin will upload all the data of the particular product in the database through a react app form and for the backend purpose we are using firebase services that gave us the option for storage and the real-time database.

With the help of storage service we were able to store the images and the 3D model at the backend and after getting it stored we will get the link for the same and then we will store the other specification and the associated link with for a specific product in the firebase real-time storage.



Now at the admin side we have given the admin the power to manipulate each and every product in the product list. we have implemented the delete and edit functionality and can only be accessed by the admin now user will be able to change the specification of any item.

We have used the CRUD functionality that for provided by the firebase cloud services through different API.



Please Enter Your Credentials

Your's Email ID:

Enter Retailer Passkey:

Enter the password

Login!!

New User!! Please [Click Here](#)

Login Page for the Retailer/ Admin



Please Enter Your Credentials

Enter your Email ID:

Enter your Name:

Enter the password

Confirm password

Sign Up!!

Already have an account!! Please [Click Here](#)

Sign Up Page for User

Passkeys are a new type of login credential that removes the need for passwords. The authentication requires either biometric authentication -- such as a fingerprint or facial recognition -- or a PIN or swipe pattern. We have used a secret key that is only available to some authorized vendor so that they can use this for the quick login and secure authentication process and easy access for the services provided for our website

FTV-Feel The Virtual

Home

Product List

Add Products

Contact Us

Register

Login ▾

Retailer


User

ENTER THE PRODUCT DETAILS:

Upload image :

Choose File


No file chosen



Upload model :

Choose File

No file chosen



Product Name:

Enter the Product Name:

Product Price:

Enter the Product Price:

Retailer Name:

Enter the Retailer Name:

Product Description:

Enter the Product Description:


Product Used for:

Wall

☐ Are you sure all the Entered Data is correct!!

submit

Add Product Page for Vendors









Please Enter Your Credentials

Your's Email ID:

Enter the password

Login!!

New User!! Please [Click Here](#)



Login Page for User

USER'S SIDE OF ECOMMERCE PLATFORM

A Normal User can first need to authenticate himself/herself in order to proceed further with exploring the website. The User and the Admin property by the simple feature that a normal user is not able to upload the product on the website as we have stated in the Admin Control above, in order to upload the product into the database we requires a special passkey which is needed to be provided in the form page.

To start with the exploring the application, he/she needed to Register themselves on the website but if they are already registered themselves earlier then they can easily login themselves. The login and the register functionality in our web application uses

the features like authentication using hashing and encryption techniques such as **salting** and including the dependencies like **uuid**. After Authenticating themselves, the user can explore different pages

(1) **Product List Page**

(2) **Display Page**

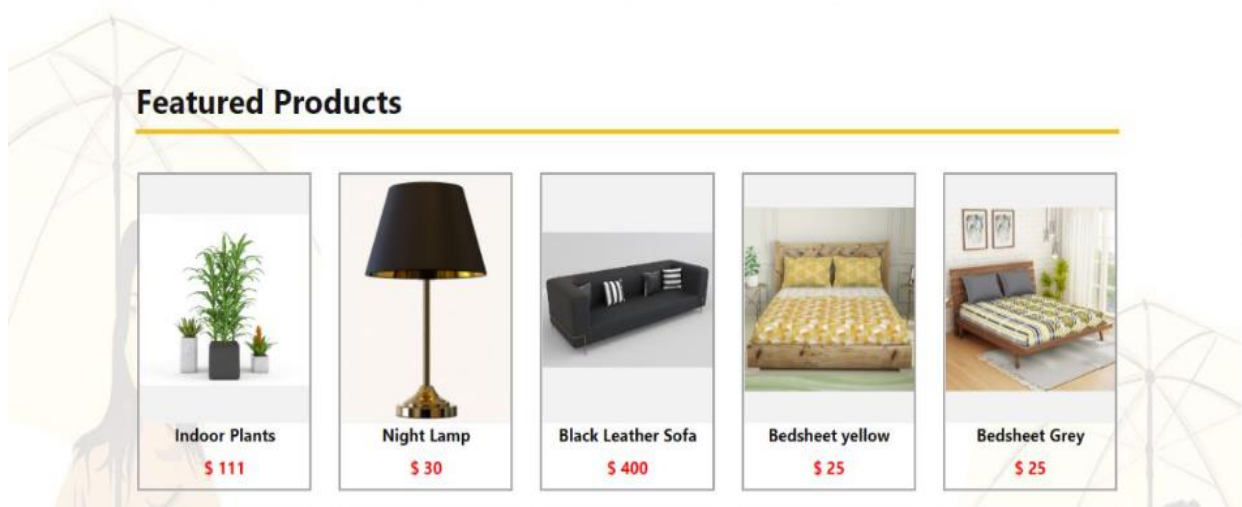
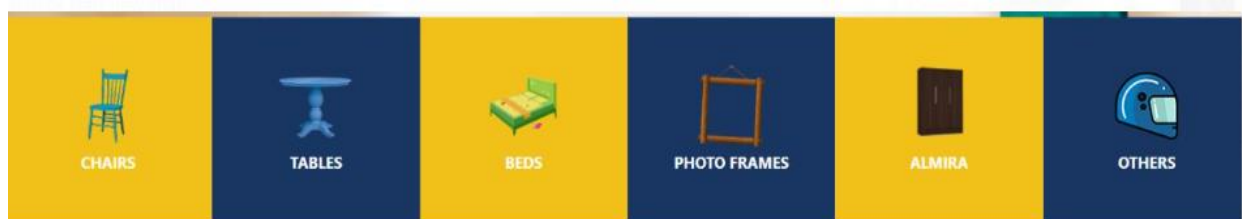
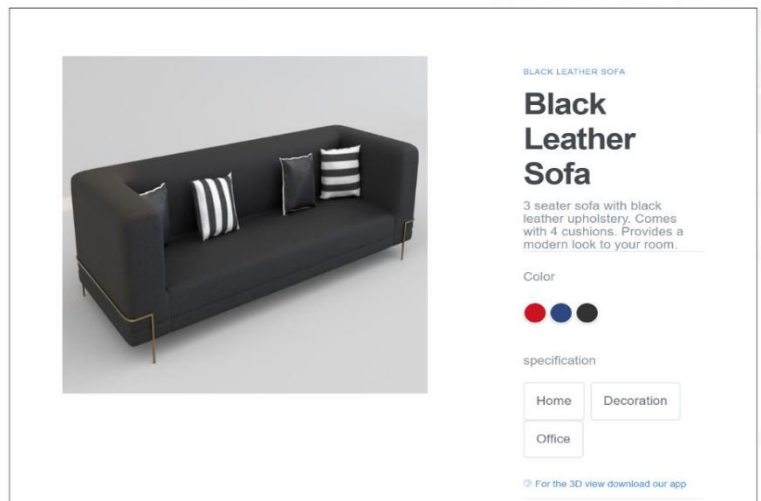
(3) **Home Page**

(4) **About Section**

(5) **Contact Us Page**

User can view all the Products that are available in the stock and their

corresponding details along with their small description. User can click on the particular product and then they are available with the product description. This is done using the dynamic routing which has been done using the React package called **React-Router-Dom** and using the **Params** functionality.



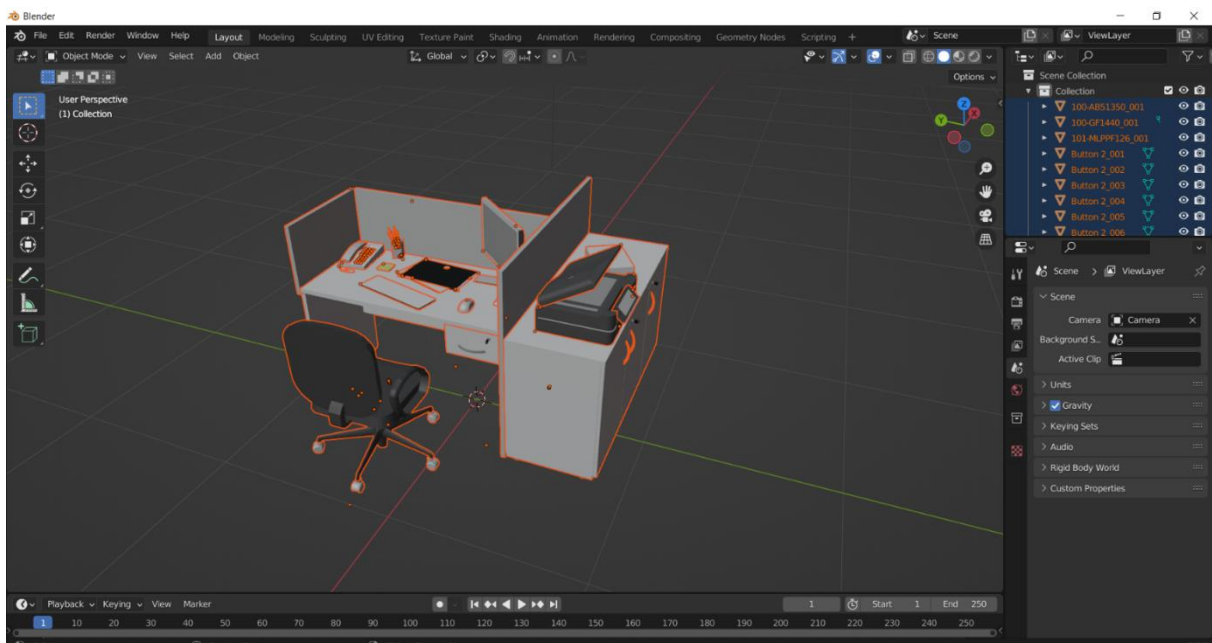
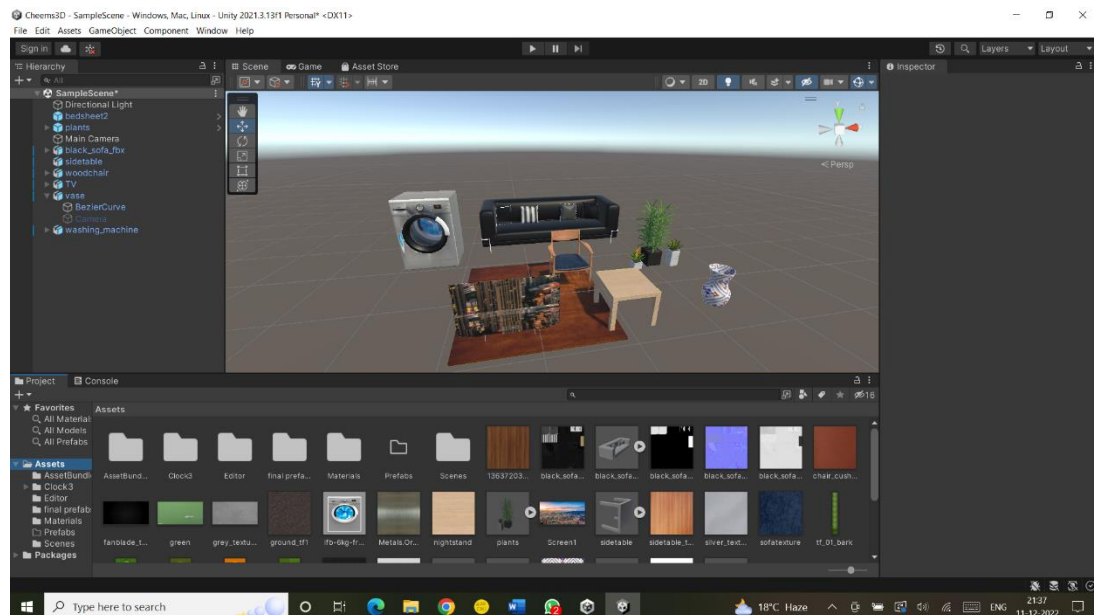
CREATING 3D ASSETS

The 3D assets needed can be created using **Unity** software. Either you can build the model up from scratch or import any 3D model having complete geometry along with its textures.

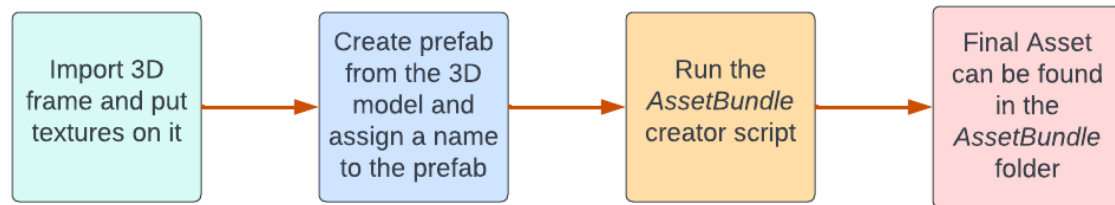
Place the textures accordingly on the model and adjust its **size, orientation and location**. Various 3D file formats supported by the Unity are:

.fbx, .obj, .dae, Autodesk Maya, 3ds Max, Blender

And the texture files that are supported are: **.png, .jpg, .tga, .mtl**



The following chart describes the steps to create the final 3D asset compatible with Android from the initial 3D model



DEVELOPING WEB BASED AR

Although we have the complete android application for the user, there is a need for a similar application for the administrator. When uploading a new model to the database, the administrator might want to see and feel how the 3D model looks in the AR.

But to test this, he/she cannot put the model on the android application as it facing the user. Therefore, a need arises for a way to check the model before deploying it on the main database.

Thus, we created the web-based AR provides basic AR functionalities for the admin but not as robust as the main AR android application.

On the right is the screenshot of the AR web page that does the following:

- Tracks and creates a local map of the surrounding by prompting the user to scan some area of the scene.
- Detects vertical and horizontal planes in the scene.

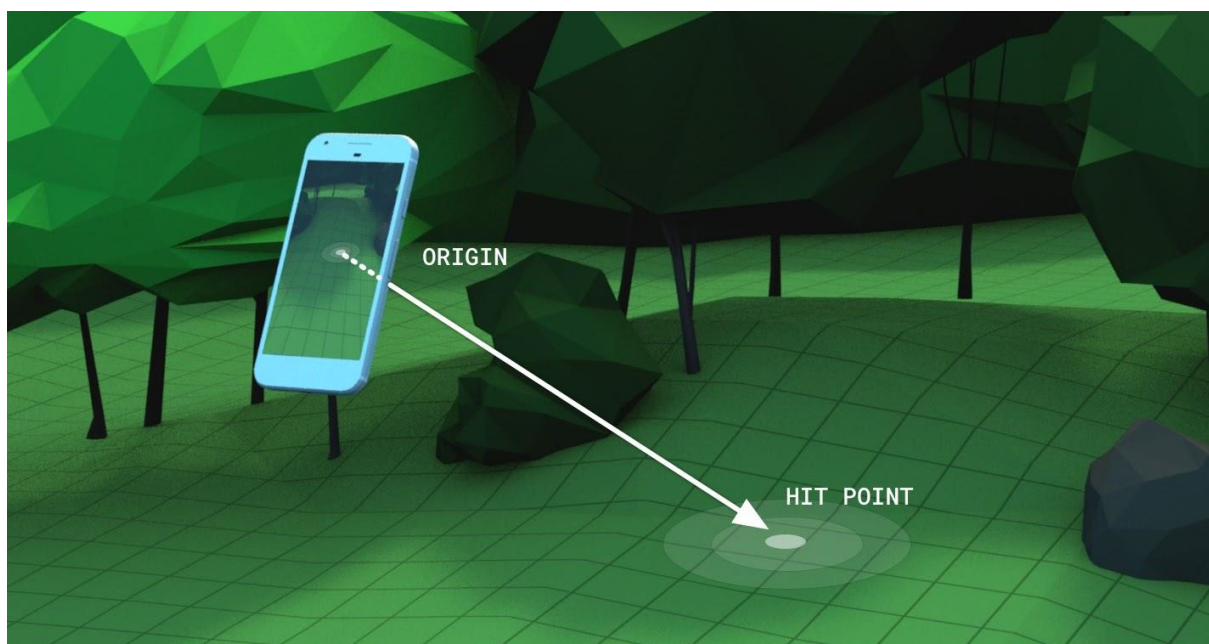


- Provides a reticle (blue target) which pin points to the most suitable location on the plane.
- The user can then put a 3D model on that target and view the model in the real world.

Understand a hit test

A hit test is generally a way to cast out a straight line from a point in space in some direction and determine if it intersects with any objects of interest. In this example, you aim the device at a location in the real world. Imagine a ray traveling from your device's camera and straight into the physical world in front of it.

The WebXR Device API lets you know if this ray intersected any objects in the real world, determined by the underlying AR capabilities and understanding of the world.



The technologies used to create it are:

1. **WebXR** – WebXR Device API is a Web application programming interface that describes support for accessing augmented reality and virtual reality devices in a web browser.
2. **THREE.JS** - Three.js is a cross-browser JavaScript library and application programming interface used to create and display animated 3D computer graphics in a web browser using WebGL. It provides us with defining, loading and managing the 3D models used in the AR.
3. **HTML, CSS, JavaScript** – We create an AR experience into a traditional webpage using existing web technologies. The HTML page is provided with CSS styling and JavaScript for enabling basic AR functionality.

The types of models supported by web-based AR are very limited:

.USDZ, .GLTF, .GLB

and do not overlap those needed by the android application. Thus, we need two separate flows for the web and android.

To setup an environment capable of AR development using WebXR, you'll need:

- A web server that serves resources in a **secure context** – either the resource is served over **TLS** (https://) or it is requested through *localhost*.
- A WebXR compatible browser on an ARCore supported device

RESULT

The following results are obtained:

1. Markerless AR technology (used in Unity) works better as compared to marker-based AR (used in Vuforia) as there is no need to track a specific object.
2. There are almost no jitters observed when using the markerless AR while the experience was not as smooth as expected in the marked-based AR.
3. AR works better in well-lit spaces.
4. It is more difficult to detect vertical surfaces than horizontal surfaces for the markerless AR.
5. The android application was made more efficient and stable than what was desired at the beginning of project with the performance exceeding the initial expectations.
6. The e-commerce platform met the expectations from both the user's side and administrator's side along with extra features than planned at the start.

CONCLUSION AND FUTURE EXTENSION

The following conclusions can be derived from the project:

1. Markerless AR performs better as it works on a generalized space mapping algorithm as compared to the marker-based AR which needs to detect specific key points in the scene.
2. As seen in marker-based AR technology (Vuforia), the target object should have strong borders/contrast between black and white colors to make tracking more stable. Smooth color transition will make recognition impossible.

Possible future work:

1. Add user authentication and payment facilities to the android AR application.
2. The web-based AR application currently works locally and independent of the ecommerce site, so integrate the application with ecommerce site to provide more comprehensive user experience.
3. Multi – Lingual functionality can be implemented in the web application which allows the users from different linguistic domain to interact with the website.
4. Cart Functionality – Introducing the cart functionality in the app allows the user to purchase more than one product at a time. Many apps nowadays provide the functionality of cart in order to maximize the user interaction.
5. Adding the Payment gateway - Adding the functionality of Purchase will introduce the new domain to our product. This payment gateway must be

built using the proper authentication and authorization in order to improve the security of the purchase.

6. Adding the Rating and Comments section in the application in order to increase the user activity and interaction. This feature helps the user in order to provide the better choice between two or more similar types of products.
7. Deploying the product to cloud, this will help to handle the excessive load that may introduce in the future with excessive user engagement and deploying more and more product in the Firebase real Time database and storage.

REFERENCES

1. <https://library.vuforia.com/>
2. <https://www.andreasjakl.com/basics-of-ar-slam-simultaneous-localization-and-mapping/>
3. <https://www.sciencedirect.com/science/article/pii/S25900005622000637>
4. <https://developers.google.com/ar/develop>
5. <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/index.html>
6. <https://developers.google.com/ar/develop/unity-arf/getting-started-ar-foundation#:~:text=Unity's%20AR%20Foundation%20is%20a,via%20Unity's%20AR%20Foundation%20package.>
7. <https://youtu.be/tBoOT9sfSug>
8. https://developer.mozilla.org/en-US/docs/Web/API/WebXR_Device_API
9. <https://developers.google.com/ar/develop/webxr/hello-webxr>
10. <https://codelabs.developers.google.com/ar-with-webxr#0>
11. <https://learn.unity.com/tutorial/introduction-to-asset-bundles>
12. <https://reactjs.org/docs/getting-started.html>
13. <https://youtu.be/6kgitEWTxac>
14. <https://youtu.be/EynAnD8nDfc>
15. <https://youtu.be/IokPiSZpXDY>
16. <https://getbootstrap.com/docs/5.0/getting-started/introduction/>