FTV: Feel The Virtual

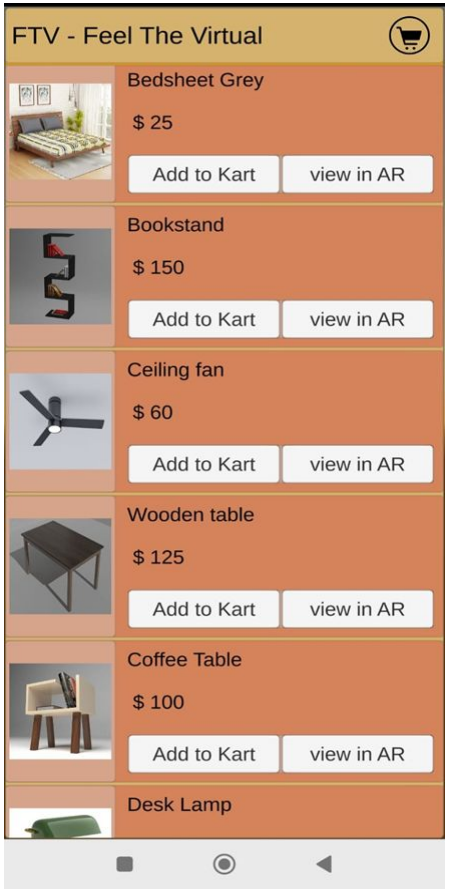
B. Tech Project Presentation
COCSC22

Shivam Sharma (2019UCO1573)
Rijul Garg (2019UCO1570)
Vishal Verma (2019UCO1563)
Shrey Rastogi (2019UCO1562)

Overview

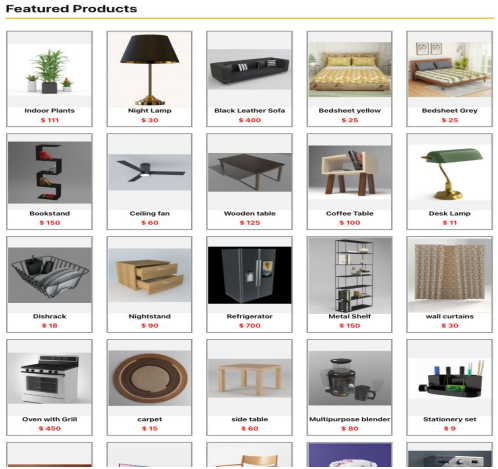
- PROBLEMS IN ONLINE SHOPPING
  - Incomplete product experience
  - Unable to assess the feel and dimensions

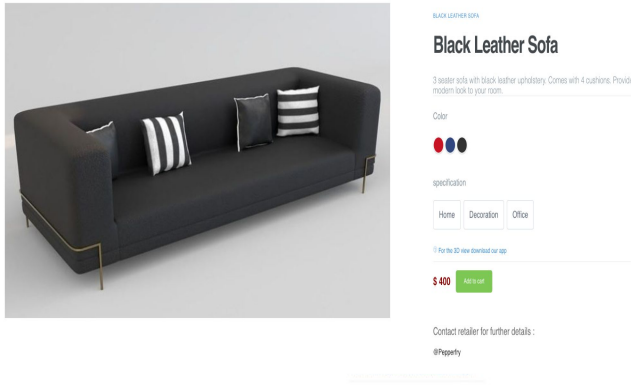
- OUR SOLUTION
  - Use AR to make product interactive
  - product visualization experience



ANDROID APPLICATION

- Preprocessing of the model before uploading
- Fetching product data and caching models
- Preprocessing models to support user interactions
- Placement of model in AR
- Interactions in AR screen





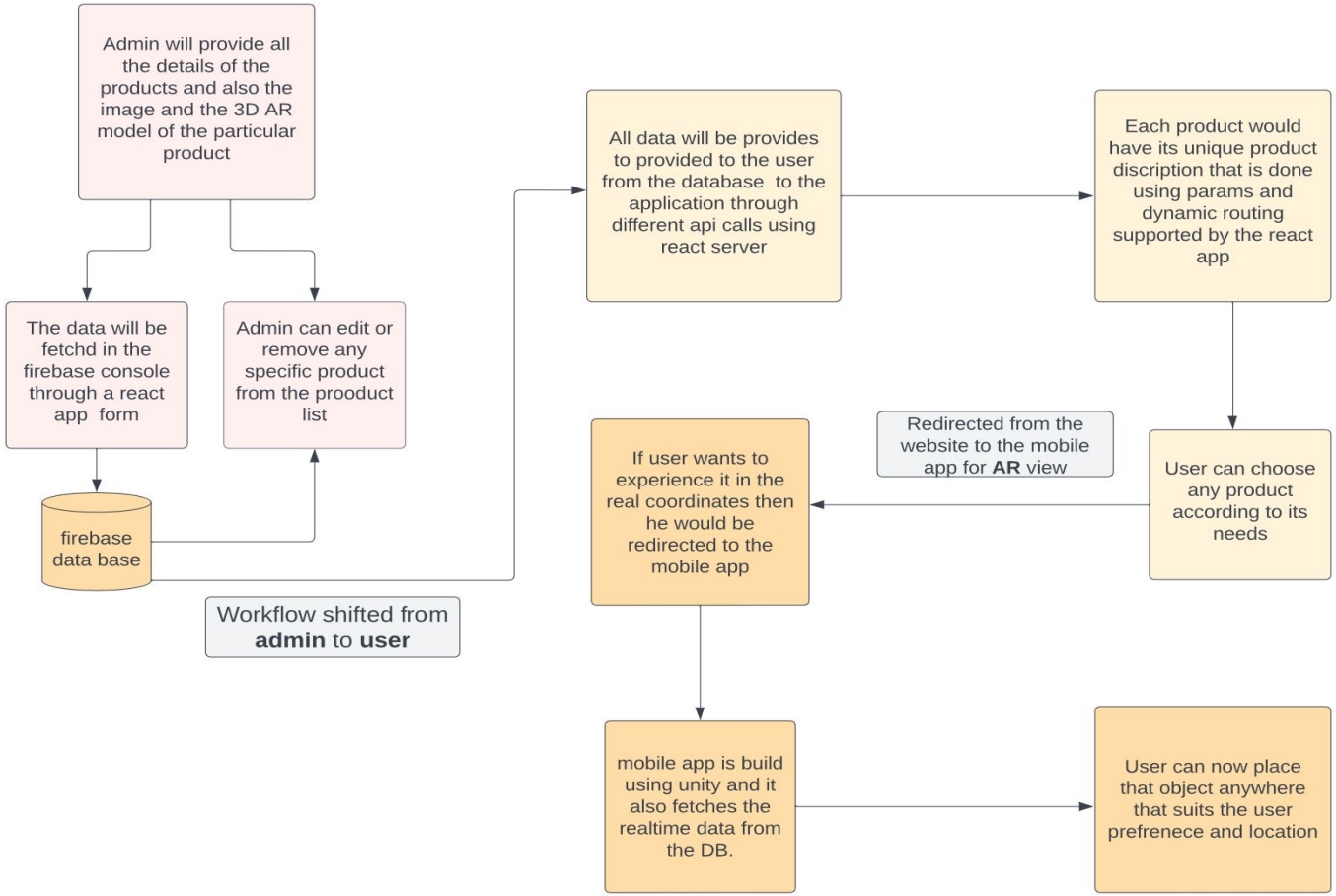
Interface for User and Admin

User Interface

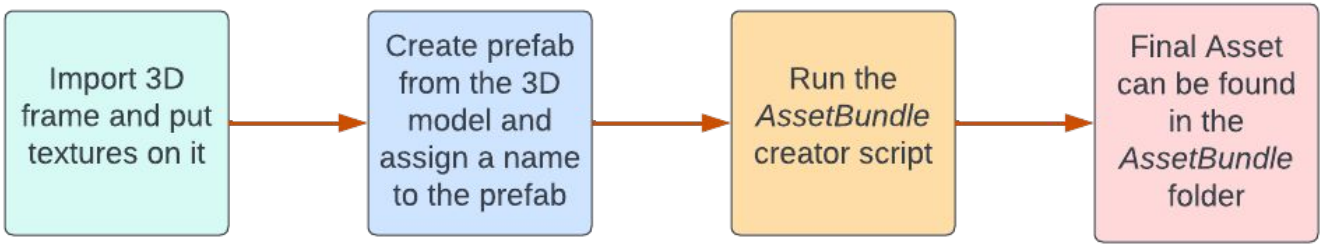
- Product List Page
- Home Page
- User Authentication
- Dynamic Routing to display specific product

Admin Interface

- Product List Page with delete control
- Authentication
- Passkey Feature
- Add Product



Flow diagram for the working of the ecommerce platform



CREATING 3D ASSETS

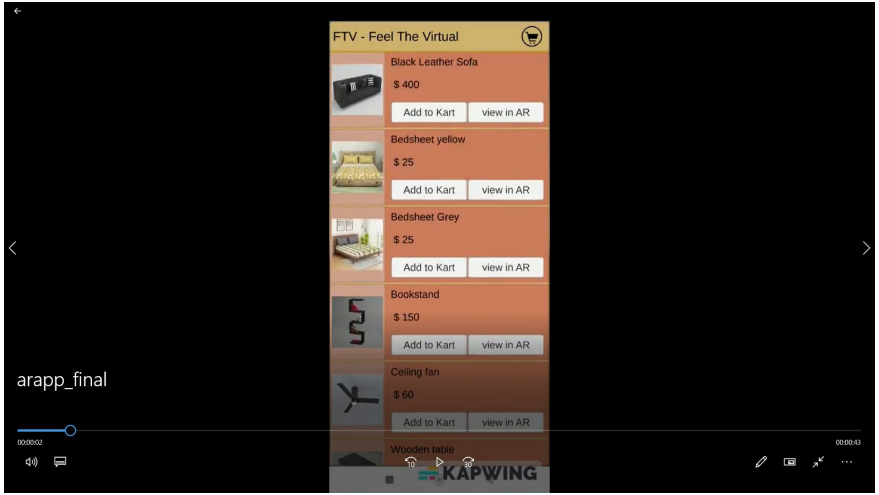
WEB-BASED AR

- ➔ For administrator to check models before uploading to website
- ➔ Made using WebXR, Three.js
- ➔ reticle (blue target) points to detected location on plane

RESULT

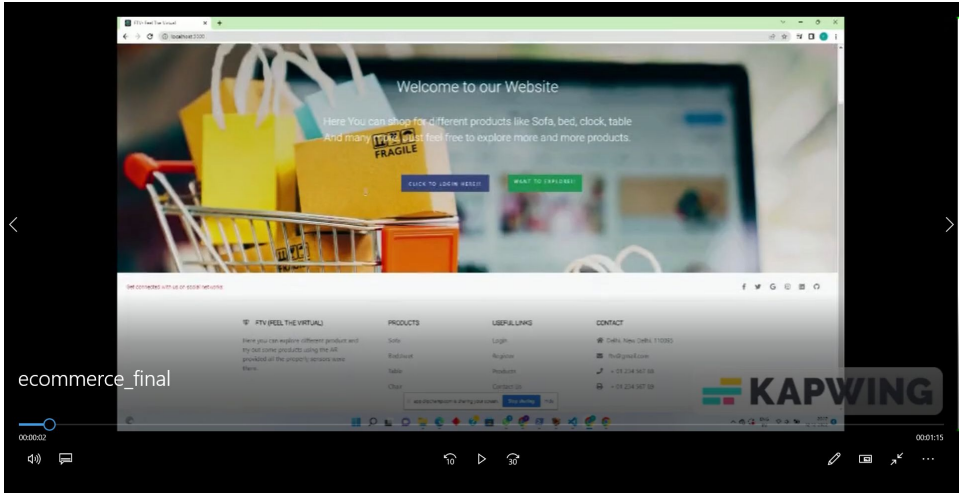
- Markerless AR is better in performance as compared to Marker-based AR.
- AR works better in well-lit spaces.
- Difficult to detect vertical planes as compared to horizontal planes.

Android Application



GOOGLE DRIVE LINK
Android Application

Ecommerce Platform



GOOGLE DRIVE LINK
Ecommerce Platform

CHALLENGES

1. Challenges with Vuforia

- **Challenge:** marker based AR, external capabilities and poor performance
- **Solution:** AR foundation, uses capabilities of device and markerless AR

1. Need for Dynamic app

- **Challenge:** local models make app massive and reduce dynamic nature.
- **Solution:** dynamic loading and assetbundle pipeline.

CHALLENGES

3. Storing the assets

- **Challenge:** Storing and retrieving images, 3D model in mongoDB
- **Solution:** Firebase allows to store all kinds of files, relatively easier to fetch data and file with realtime DB.

4. Update the state

- **Challenge:** Needed an engine that dynamically render the page with change in DB state.
- **Solution:** Simple views for each state in the application, React will efficiently update and render the components when data changes .Firebase CRUD operation were comparatively much feasible and easier.

REFERENCES

1. https://www.andreasjakl.com/basics-of-ar-slam-simultaneous-localizationand-mapping/
2. https://www.sciencedirect.com/science/article/pii/S2590005622000637
3. https://developers.google.com/ar/develop
4. https://reactjs.org/docs/getting-started.html
5. https://learn.unity.com/tutorial/introduction-to-asset-bundles
6. https://developers.google.com/ar/develop/webxr/hello-webxr
7. https://developer.mozilla.org/en-US/docs/Web/API/WebXR_Device_API
8. https://library.vuforia.com/
9. https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual /index.html
10. https://codelabs.developers.google.com/ar-with-webxr#0
11. https://getbootstrap.com/docs/5.0/getting-started/introduction/
12. https://youtu.be/LO4YTml3IAQ
13. https://youtu.be/6kgitEWTxac