

**A
Project Report
on
3D MODEL MARKETPLACE
BTech-IT, Sem VI**

Prepared By:
IT-125 Rathva Dharmendra

Guided By:
Prof. Kunal J. Sahitya
Dept. of Information Technology



Department of Information Technology
**Faculty of technology,
Dharmsinh Desai University
College road, Nadiad- 387001**

April, 2025

CANDIDATE’S DECLARATION

We declare that 6th semester report entitled “3D Model Marketplace” is my own work conducted under the supervision of the guide Prof. Kunal J. Sahitya.

We further declare that to the best of our knowledge the report for B.Tech. VI semester does not contain part of the work which has been submitted either in this or any other university without proper citation.

Candidate’s Signature
IT125 RATHVA DHARMENDRA
Id: 22itutf046

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT



CERTIFICATE

This is to certify that the project carried out in the subject of Project-I, entitled
“3D Model Marketplace” and recorded in this report is a bonafide report of work of
1)Rathva Dharmendra Roll No. IT125 ID No: 22ITUTF046

of Department of Information Technology, semester VI. we were involved in Project
work during academic year 2024 -2025

Prof. Kunal J. Sahitya
(Project Guide),
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. (Dr.) V. K. Dabhi
Head , Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the Department of Information Technology, Dharmsinh Desai University, for providing us with the opportunity to work on this project. We extend our heartfelt thanks to Prof. Kunal j. sahitya, our project guide, for his invaluable guidance, support, and encouragement throughout the project.

We also thank all faculty members and staff of the **Information Technology Department** for their continuous support and knowledge, which have helped us in successfully completing this project. Lastly, we acknowledge the contribution of our peers and classmates who provided insightful feedback and motivation during this journey.

Rathva Dharmendra (IT125)

TABLE OF CONTENTS

No.	Title	Pg. No.
1.	Abstract	vii
2.	List of Figures	viii
3.	List of Tables	ix
4.	Chapter 1: Introduction	1
5.	Chapter 2: Project Management	2
6.	Chapter 3: System Requirements Study	5
7.	Chapter 4: System Analysis	8
8.	Chapter 5: System Design	14
9.	Chapter 6: Implementation Environment	18
10.	Chapter 7: Testing	19
11.	Chapter 8: User Manual	21
12.	Chapter 9: Limitations and Future enhancements	27
13.	Chapter 10: Conclusion and Discussion	28
14.	References	30

DETAILS OF CHAPTERS

1. Introduction.....	1
1.1 Project Details.....	1
1.2 Purpose.....	1
1.3 Scope.....	1
1.4 Objective.....	1
1.5 Technology Review.....	1
2 Project Management.....	2
2.1 Feasibility Study	
2.1.1 Technical feasibility.....	2
2.1.2 Time Schedule feasibility.....	2
2.1.3 Operational feasibility.....	2
2.1.4 Implementation feasibility.....	2
2.2 Project Planning.....	3
2.2.1 Project Development Approach and Justification.....	3
2.2.2 Project Plan.....	3
2.2.3 Milestones and Deliverables.....	3
2.2.4 Roles and Responsibilities.....	3
2.2.5 Group Dependencies.....	4
2.3 Project Scheduling.....	4
Project Scheduling Chart.....	4
3 System Requirements Study.....	5
3.1 Study of Current System.....	5
3.2 Problems and Weaknesses of Current System.....	5
3.3 User Characteristics.....	5
3.4 Hardware and Software Requirements.....	5
3.5 Constraints.....	6
3.5.1 Regulatory Policies.....	6
3.5.2 Hardware Limitations.....	6
3.5.3 Interfaces to Other Applications.....	6
3.5.4 Parallel Operations.....	6
3.5.5 Higher Order Language Requirements.....	6
3.5.6 Reliability Requirements.....	6
3.5.7 Criticality of the Application.....	6
3.5.8 Safety and Security Consideration.....	6
3.6 Assumptions and Dependencies.....	7

4	System Analysis.....	8
4.1	Requirements of New System.....	8
4.1.1	User Requirements.....	8
4.1.2	System Requirements.....	8
4.2	Features Of New System.....	8
4.3	Navigation Chart.....	9
4.4	Data Modelling.....	10
4.4.1	Data Dictionary.....	10
4.4.2	ER Diagram.....	13
5	System Design.....	14
5.1	Use Case Diagram.....	14
5.2	Class Diagram.....	15
5.3	Sequence Diagram.....	16
5.4	Activity Diagram.....	17
6	Implementation Planning.....	18
6.1	Implementation Environment.....	18
6.2	Program/Modules Specification.....	18
6.3	Coding Standards.....	18
7	Testing.....	19
7.1	Testing Plan.....	19
7.3	Testing Strategy.....	19
7.3	Testing Methods.....	19
7.4	Test Cases.....	19
7.4.1	Purpose.....	19
7.4.2	Required Input.....	19
7.4.3	Expected Result.....	19
7.4.4	Testing Types.....	20
8	User Manual.....	21
8.1	Getting Started.....	21
8.2	Features & Navigation.....	21
8.3	Troubleshooting & Support.....	21
9	Limitation and Future Enhancement.....	27
10	Conclusion and Discussion.....	28
10.1	Conclusions.....	28
10.2	Discussion.....	28
10.2.1	Self-Analysis of Project Viabilities.....	28
10.2.2	Problem Encountered and Possible Solutions.....	28
10.2.3	Summary of Project work.....	29
	REFERENCES.....	30

ABSTRACT

The **3D Model Marketplace** is a full-stack web application developed using the MERN stack to streamline the management and distribution of 3D model assets. This platform enables administrators to upload and organize models into well-defined categories and subcategories, while users can browse, filter, and download them effortlessly. It features secure user authentication, ZIP file storage, and intuitive content organization. The user interface is designed for easy navigation and interaction. Admins are empowered with full CRUD capabilities to manage models and platform content efficiently.

The project aims to improve the accessibility, structure, and controlled distribution of 3D models in a scalable and modern environment. It also lays a strong foundation for future features, including licensing, advanced search, secure payment integration, and user-uploaded content for model selling. This makes it a powerful and adaptable tool for digital content distribution in the 3D design space.

LIST OF FIGURES

	Name	Page No.
Fig 2.1	Agile Model	3
Fig 4.1	Navigation Chart for 3D Model Market- place System Development	9
Fig 4.2	Navigation Chart of 3D Model Marketplace	9
Fig 4.3	ER Diagram	13
Fig 5.1	Use Case Diagram	14
Fig 5.2	Class Diagram	15
Fig 5.3	Seq. Diagram	16
Fig 5.4	Activity Diagram	17

LIST OF TABLES

	Name	Page No.
2.3	Project scheduling	4
4.4.1 - 1	User	10
4.4.1 - 2	Product	10
4.4.1 - 3	Category	11
4.4.1 - 4	Sub Category	11
4.4.1 - 5	Cart Product	12
7.4.3	Testing Types	20
10.2.2	Problem - Solution	27

CHAPTER 1: INTRODUCTION

1.1 Problem Statement

This project is a web application where users can buy and download 3D model ZIP files. It supports both free and paid models, and provides categories and subcategories for easy searching. The platform allows admins to upload models, manage them, and users to download files after login or payment.

1.2 Proposed Solution

The main purpose of **3D Model Marketplace** is to create an online platform for sharing and selling 3D models in the form of ZIP files. It helps 3D artists or creators to upload their work and get buyers or users who are interested in 3D assets.

1.3 Scope

- Users can register and log in.
- Admins can manage categories, subcategories, and 3D model uploads.
- Users can view and download models after login or payment.
- Models are Secure stored on API, and data is saved in Database.
- The site has a clean and modern UI using Tailwind CSS.

1.4 Objectives

- Build a platform for 3D model sharing.
- Enable secure file uploads and downloads.
- Classify 3D models into structured categories and subcategories for organized browsing.
- Provide a user-friendly interface for both users and admins.

1.5 Technology Review

- Frontend: React.js with Tailwind CSS for an interactive and responsive UI.
- Backend: Node.js + Express.js.
- Database: MongoDB for Large and Complex Data Management.
- File Storage: Cloudinary (Store Zips in raw Folder).
- Tools/Libraries: Axios, React Router, JWT, CORS, dotenv, cookie-parser, Morgan, Helmet.

CHAPTER 2: Project Management

2.1 Feasibility Study:

2.1.1 Technical Feasibility

The project is technically possible using the MERN stack (MongoDB, Express, React, Node.js). File Storage API is used for ZIP file uploads, and all tools used are compatible with web-based platforms. All chosen technologies are open-source and widely supported, ensuring future scalability. The system architecture supports modular development, making it easy to maintain and extend.

2.1.2 Time Schedule Feasibility

The project was completed in a reasonable time frame with proper planning, covering user login, admin panel, file upload, download system, and category and sub category management spanning 3-4 months.

2.1.3 Operational Feasibility

The system is user-friendly and can be operated by both technical and non-technical users. Admin features are well-structured, making it easy to manage the platform.

2.1.4 Implementation Feasibility

The implementation was smooth using modular coding. Frontend and backend are separated, and APIs are properly connected with security in place.

2.2 Project Planning

2.2.1 Project Development Approach and Justification

The agile development approach is used, allowing for flexibility, iterative improvements, and continuous feedback. This ensures the system meets evolving user requirements.



Fig 2.1

2.2.2 Project Plan

The development process consists of:

- Requirement Analysis: Understanding system needs.
- Design & Prototyping: Create UI mockups and define database schema for users, models, and categories and subcategories.
- Development: Build APIs with Express, integrate React frontend, and configure API for ZIP uploads.
- Testing: Unit testing, integration testing, and user acceptance testing.
- Deployment & Maintenance: Hosting the application and providing support.

2.2.3 Milestones and Deliverables

- Week 1-2: Requirement Analysis & System Design.
- Week 3-5: Backend and Database Development.
- Week 6-8: Frontend Development and Integration.
- Week 9-10: Testing and Debugging
- Week 11-12: Deployment and Final Review.

2.2.4 Roles and Responsibilities

- User: Can register, log in, browse/download models, and manage their profile.
- Admin: Manages users, categories, subcategories, model uploads, and overall platform operations.

2.2.5 Group Dependencies

As this was a solo project, dependencies were minimal. Each module was built to work independently and connected using APIs.

2.3 Project Scheduling

A detailed timeline with sprints and deadlines ensures timely completion of the project. Tasks are completed by alone me only to maintain a structured and efficient workflow.

Task	Week 1-2	Week 3	Week 4-6	Week 6-8	Week 8-9	Week 10-11	Week 12-13	Week 14
Requirement Analysis	Done							
UI/UX Design		Done						
Frontend Development			Done	Done	Done	Done	Done	
Backend Development			Done	Done	Done	Done	Done	
Database Setup			Done	Done	Done			
Security Implementation					Done	Done		
Testing						Done	Done	
Deployment							Done	Done

CHAPTER 3: SYSTEM REQUIREMENTS STUDY

3.1 Study of Current System

There is no existing dedicated system for this specific project. Currently, 3D model sharing is done through third-party platforms, which often lack support for ZIP-based downloads, custom categories, subcategories, and admin-level content management.

3.2 Problems and Weaknesses of the Current System

- Difficult to find categorized and sub categorized 3D models.
- Lack of Automation, No direct download option after login/payment
- Manual upload process is slow.
- Limited admin control in some existing platforms.

3.3 User Characteristics

- Admin: 3D models, categories, and subcategories. Oversees user activity and ensures smooth operation of the platform.
- User: Can register, log in, view models, and download files.
- Both types of users can use the system easily due to its simple design.

3.4 Hardware and Software Requirements

Hardware Requirements

- Processor: Minimum Intel Core i3
- RAM: 4GB or higher
- Storage: 100GB or more
- Stable internet connection

Software Requirements

- Operating System: Windows/Linux
- Browser: Chrome, Firefox, Edge
- File Storage: Cloudinary Account
- Other Tools: Postman (API testing), GitHub (version control), VS Code.

3.5 Constraints

3.5.1 Regulatory Policies

All models must follow copyright laws. Only original or licensed files can be uploaded. Only original content or properly licensed files are allowed on the platform to avoid legal issues.

3.5.2 Hardware Limitations

Limited system storage and processing power may affect the performance of large file uploads. Users with slower internet or outdated hardware may experience delays during upload or download operations.

3.5.3 Interfaces to Other Applications

Files Storage API Used for uploading, storing, and retrieving ZIP files of 3D models securely and MongoDB Serves as the primary database for storing user data, model metadata, categories, and subcategories.

3.5.4 Parallel Operations

Multiple users (faculty and students) can interact with the system simultaneously without performance degradation.

3.5.5 Higher Order Language Requirements

The project is developed using JavaScript for both the frontend and backend. React.js is used to create a dynamic and responsive user interface, while Node.js with Express handles the backend logic and API integration.

3.5.6 Reliability Requirements

The system must be highly reliable, ensuring data integrity and secure storage and Uses JWT for authentication and secure file access.

3.5.7 Criticality of the Application

The application is important for users who require quick, secure, and reliable access to 3D models for design, development, or creative purposes.

3.5.8 Safety and Security Consideration

- Authentication: Uses JWT for authentication and secure file access.
- Data Encryption: Secure storage of 3D Model Zip Files data and user information.
- Error Handling: Proper error handling and API protection.

3.6 Assumptions and Dependencies

- A reliable internet connection is required for users to access, upload, and download 3D models effectively.
- API services will remain available and responsive for file uploads and retrieval.
- Database will maintain a stable connection for smooth data storage and access.

CHAPTER 4: SYSTEM ANALYSIS

4.1 Requirements of the New System

4.1.1 User Requirements

- Users need a platform where they can easily browse, search, and download 3D model ZIP files. They expect clear model categories, fast download options, and a simple login system.

4.1.2 System Requirements

- User registration and login.
- Admin control panel to manage uploads.
- Category and subcategory management.
- Cloud-based ZIP file upload and download.
- A clean user interface with fast loading pages.

4.2 Features of the New System

- Users can view free or paid models.
- Admins can upload ZIP files with model details and thumbnails.
- Each model is grouped into categories and subcategories.
- After login (and payment if needed), users can download ZIP files.
- Uses Secure File Storage to store files and MongoDB to store file data.
- Search and filter features to quickly find models.

4.3 Navigation Chart

4.3.1 Navigation Chart For 3D Model Marketplace System Development

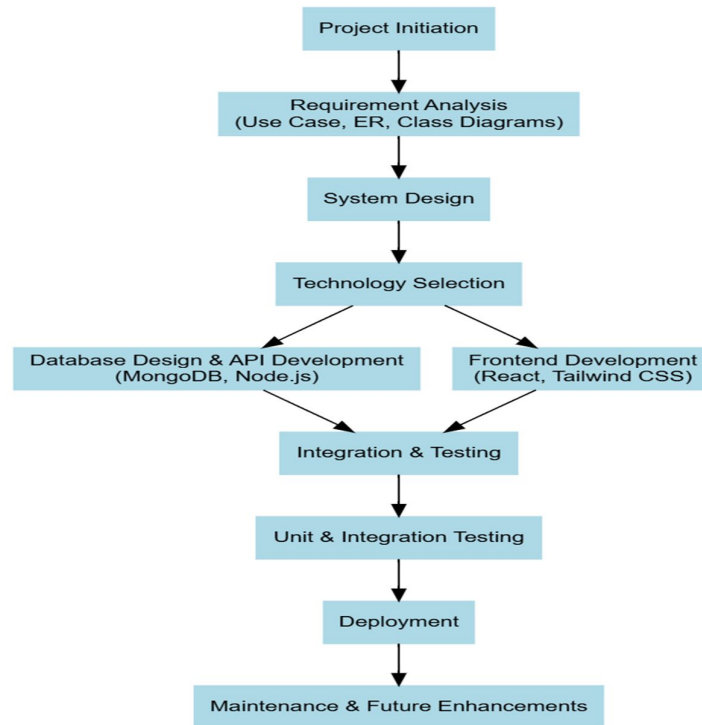


Figure 4.1 Navigation Chart of Development

4.3.2 Navigation Chart For 3D Model Marketplace

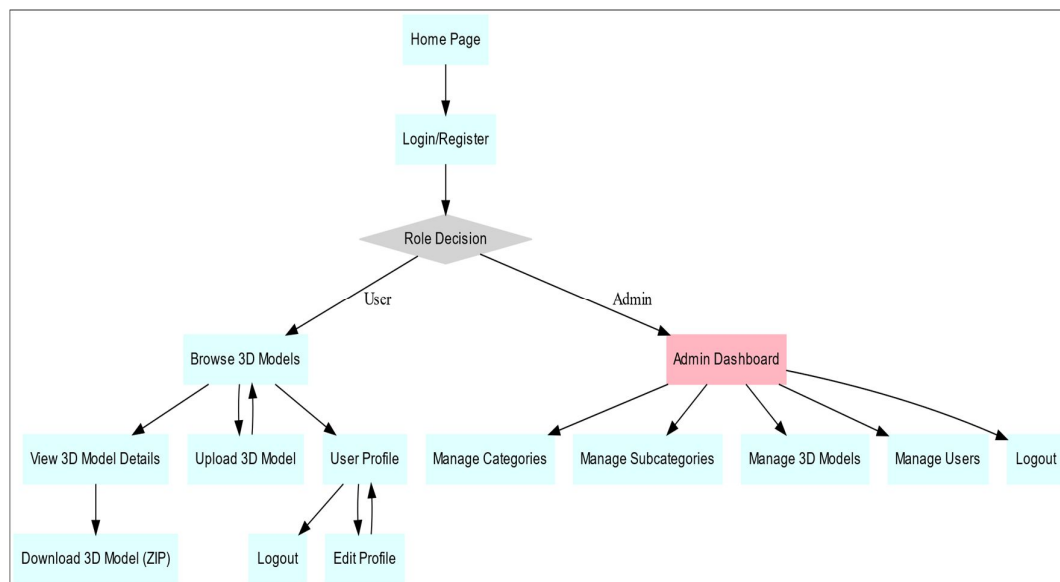


Figure 4.2 Navigation Chart of System

4.4 Data Modelling

4.4.1 Data Dictionary

1. User Table

Column Name	Data Type	Description
name	String	The user's full name.
email	String	The user's unique email address.
password	String	User's password.
avatar	String	URL of the user's avatar image.
mobile	Number	Mobile number.
last_login_date	Date	Last login date.
role	String	Either admin or user.
status	String	Active, Inactive, or Suspended.

Relationships:

- One-to-Many with Cart Product.

2. Product Table

Column Name	Data Type	Description
_id	ObjectId	Primary Key, Unique product ID
name	String	Product name
image	Array	Product images
category	ObjectId[]	Foreign Key, References to Category Model
SubCategory	ObjectId[]	Foreign Key, References to Sub Category Model
extension	String	File extension (.max, .blend, etc.)
price	Number	Product price
discount	Number	Discount on product
description	String	Product description
more_details	Object	Key-value info
publish	Boolean	Publish status
zipFile	Object	Object containing zip file info
createdAt	Date	Auto timestamp

Relationships:

- Many-to-Many with Category.
- Many-to-Many with Sub Category.

3. Category Table

Column Name	Data Type	Description
_id	ObjectId	Primary Key, Unique ID for category
name	String	Category name
image	String	Category image URL
createdAt	Date	Auto timestamp
updatedAt	Date	Auto timestamp

Relationships:

- One-to-Many with Sub Category.
- Many-to-Many with Sub Category (Many-to-Many possibility due to array).

4. Sub Category Table

Column Name	Data Type	Description
_id	ObjectId	Primary Key, Unique ID for Sub category
name	String	Sub Category name
image	String	Category image URL
category	ObjectId[]	References Category Model
createdAt	Date	Auto timestamp
updatedAt	Date	Auto timestamp

Relationships:

- Many-to-Many with Product.

5. Cart Product Table

Column Name	Data Type	Description
_id	ObjectId	Primary Key, Unique ID for category
productId	String	Category name
userId	String	Category image URL
createdAt	Date	Auto timestamp
updatedAt	Date	Auto timestamp

Relationships:

- Many-to-One with Product.

4.4.2 ER Diagram

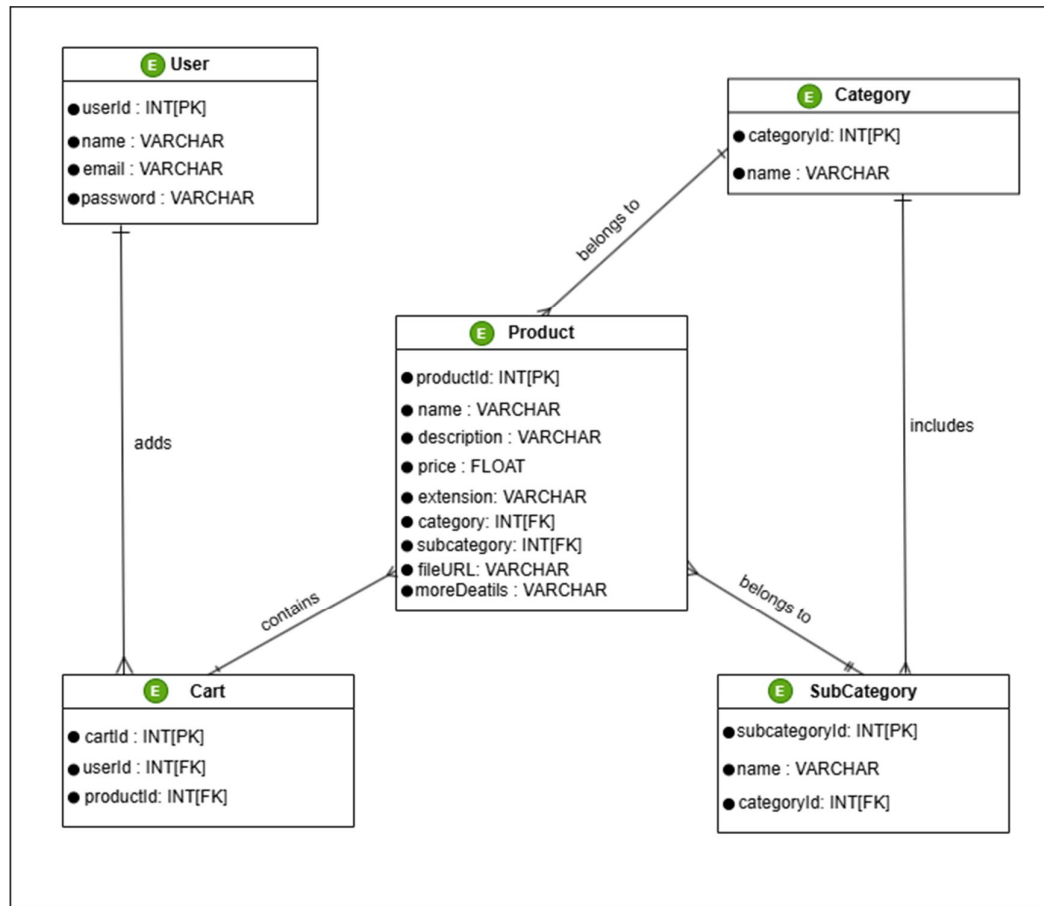


Fig 4.3 ER Diagram

CHAPTER 5: SYSTEM DESIGN

5.1 Use Case Diagram



Figure 5.1 Use Case Diagram

5.2 Class Diagram

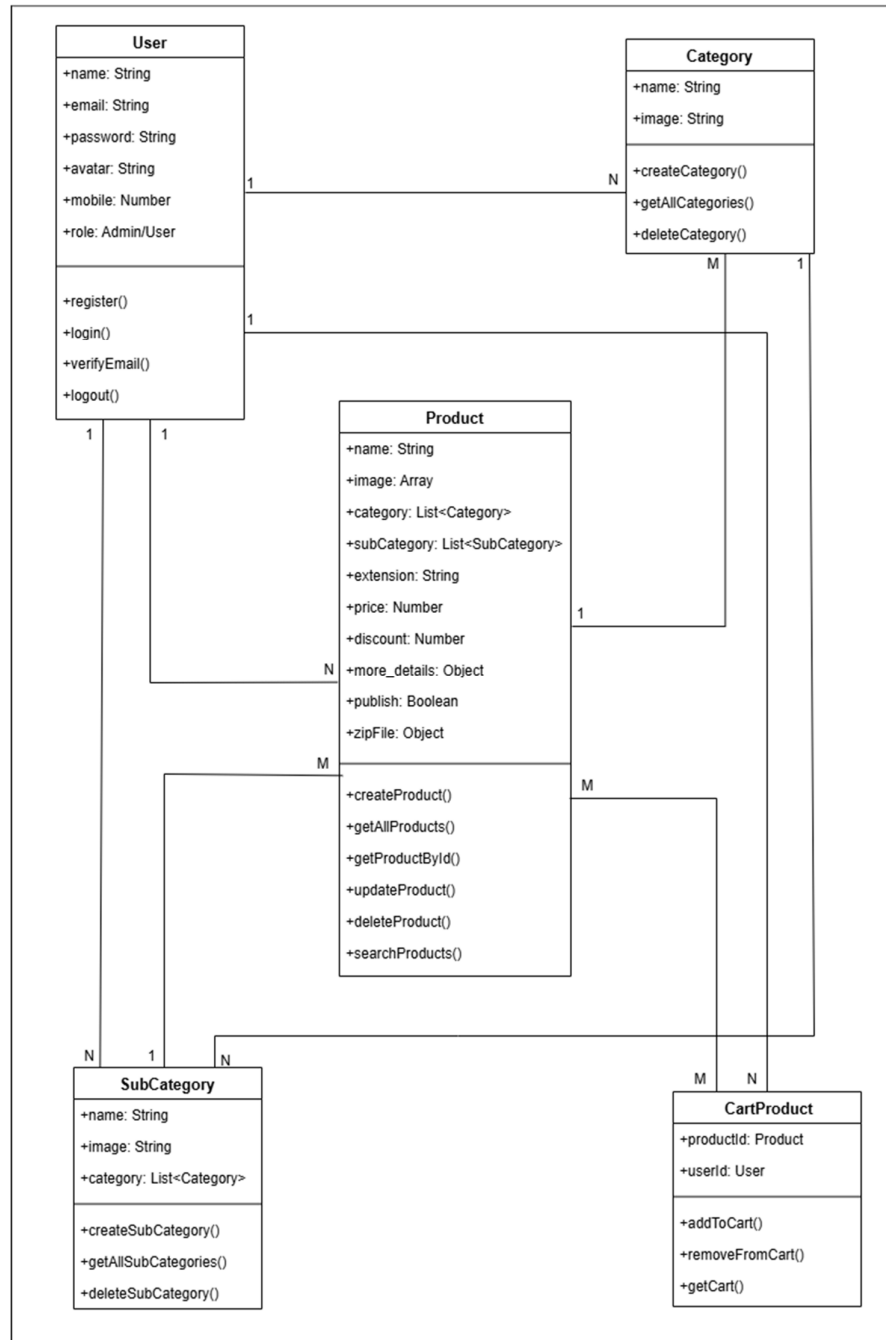


Figure 5.2 Class Diagram

5.3 Sequence Diagram

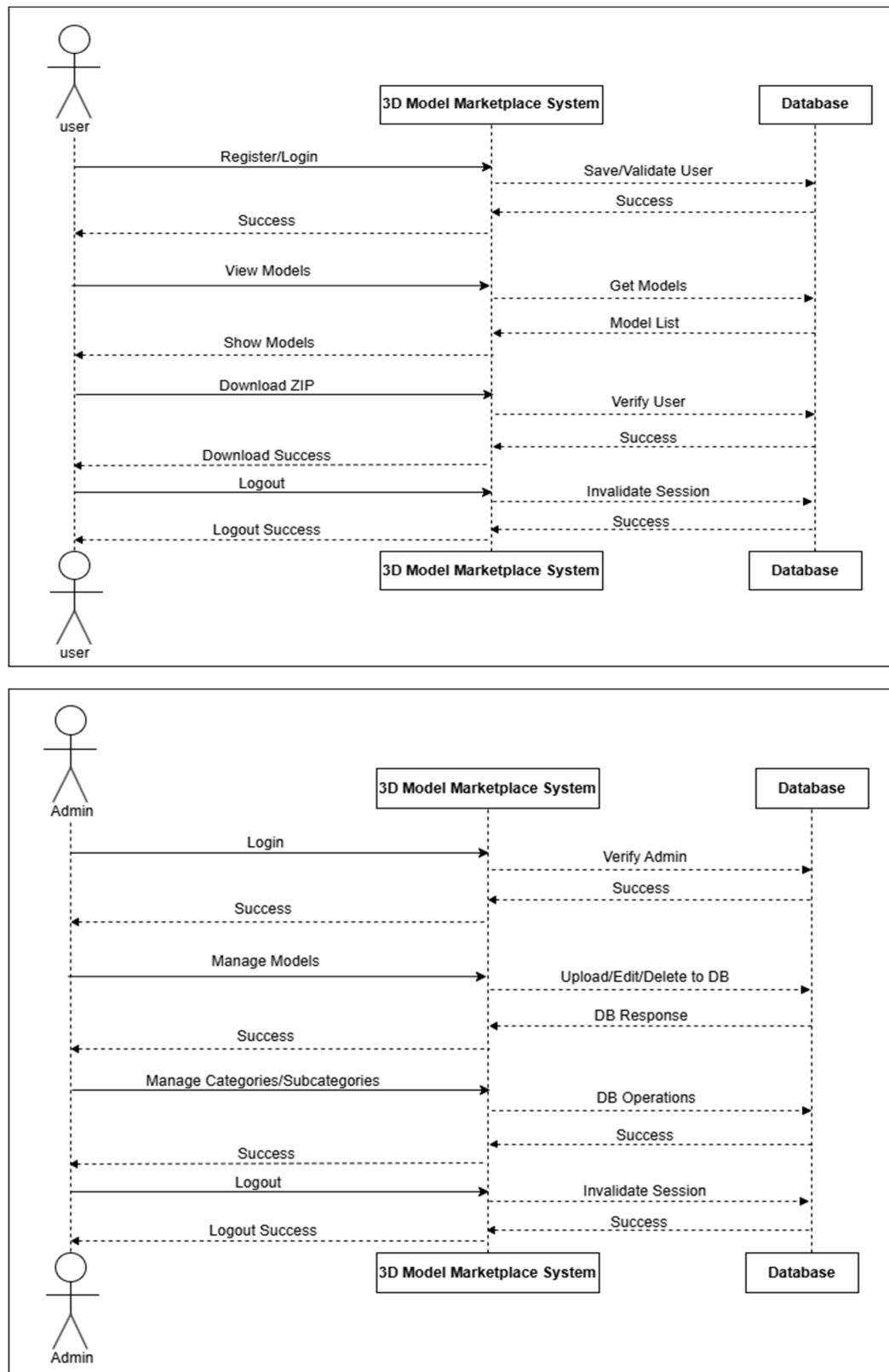


Figure 5.3 Sequence Diagram

5.4 Activity Diagram

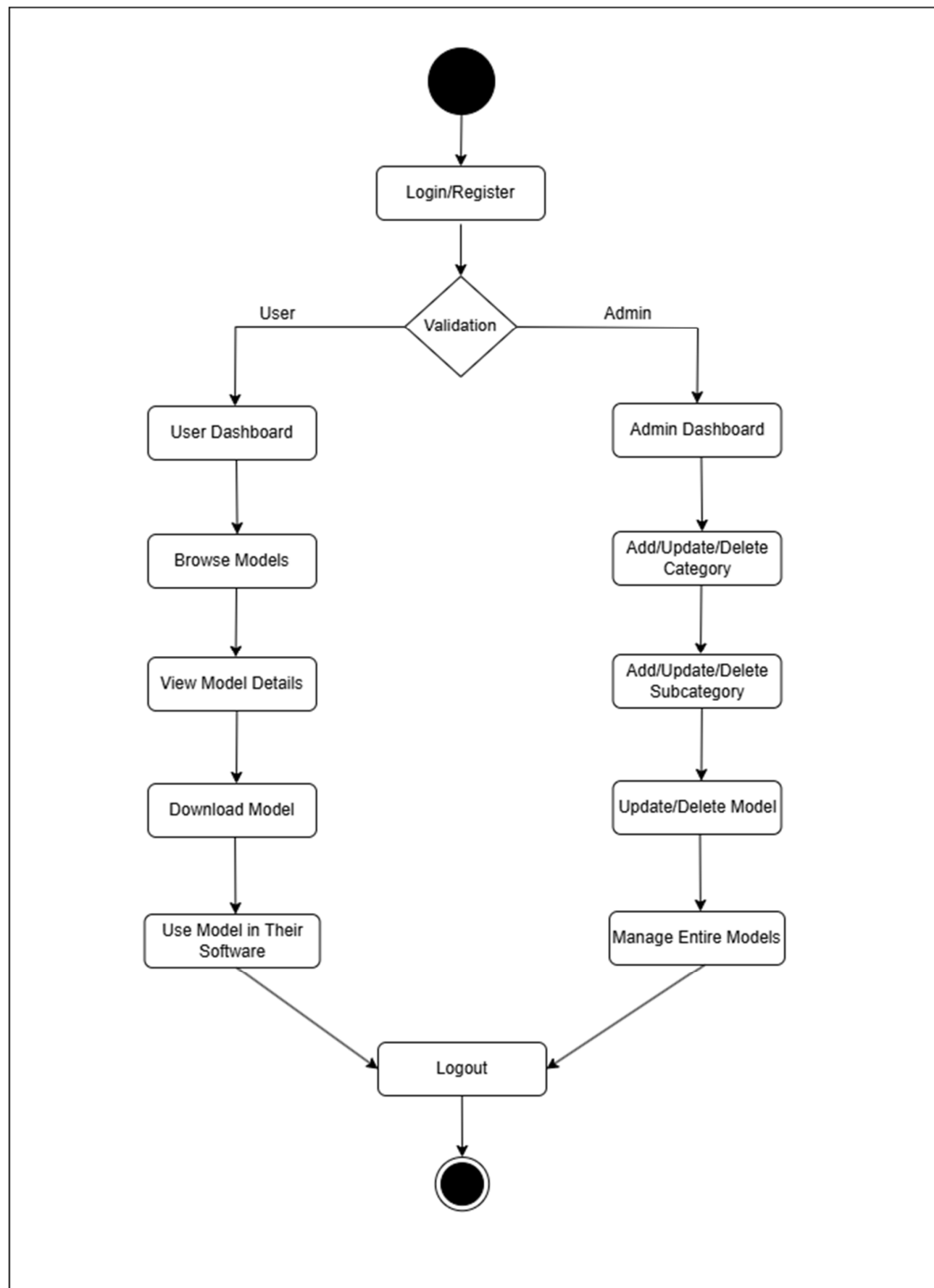


Figure 5.4 Activity Diagram

CHAPTER 6: IMPLEMENTATION PLANNING

6.1 Implementation Environment

To build the 3D model sharing website, the following tools and technologies were used:

- Backend: Node.js with Express.js
- Frontend: React.js with Tailwind CSS
- Database: MongoDB (NoSQL)
- Authentication: JSON Web Token (JWT)

6.2 Program/Modules Specification

- Authentication Module: User registration and login using JWT tokens.
- Admin Panel Module: Upload models (ZIP), create categories and subcategories.
- Category Management Module: Create and display categories and subcategories.
- Model Upload & Display Module: Upload ZIP files with preview image and info and Display models under the correct category/subcategory.
- Payment & Download Module (Optional for future integration): For now, free models are directly downloadable and Paid model download logic is ready for payment integration.

6.3 Coding Standards

- File names use lowercase and hyphen or camelCase (modelUpload.js, authRoutes.js)
- Folders are separated logically (/routes, /models, /controllers)
- Used consistent comments to explain code.
- Ensured all sensitive data like Cloudinary keys and JWT secret key are stored in .env file.

CHAPTER 7: TESTING

7.1 Testing Plan

- Unit Testing (Backend APIs)
- UI Testing (Frontend)
- Integration Testing (API & Database)

7.2 Testing Strategy

- Manual & Automated Testing for UI and API performance.
- Security Testing for authentication.

7.3 Testing Methods

- White Box Testing: Internal logic validation.
- Black Box Testing: Functional behavior testing.

7.4 Test Cases

7.4.1 Purpose

To verify each feature works as expected (login, upload, category filter, subcategory filter, download, etc.).

7.4.2 Required Input

- To Email and password for login.
- ZIP file, model name, category, subcategory and Other Details for upload.
- Model selection by user for download

7.4.3 Expected Result

- User logs in successfully
- Admin uploads model and sees it listed
- Users view and download the correct ZIP file
- No broken links or missing data

7.4.4 Testing Types:

Testing Type	Purpose
Unit Testing	Validates individual components (functions, classes).
Integration Testing	Checks interactions between modules (e.g., API & database).
System Testing	Checks that the whole webapp works start to finish.
<i>User Acceptance Testing (UAT)</i>	Ensures real-world usability before deployment.

CHAPTER 8: USER MANUAL

8.1 Getting Started

- **Login/Register:** Go to the website, click Register, and sign up using your name, email, and password.
- If already registered, click Login, enter your email and password and submit.

8.2 Features & Navigation

For Admin:

- **Login:** Access the admin dashboard after login.
- **Upload a 3D Model:** Go to the Upload Section, enter the Model name, description, choose a category and subcategory, upload the ZIP file, and submit the form to upload the model to the system.
- **Manage Categories and Subcategories:** Add or remove categories/subcategories based on model types.
- **View and Manage Models:** See all uploaded models, and edit or delete them as needed to keep the system updated.

For Users:

- **Login:** Access the user dashboard after login.
- **Browse 3D Models:** After login, go to the homepage. You can either use the search bar to find specific models or select a category and then a subcategory to browse models based on your selection.
- **View Model Details:** Click on any model to view its full details, including pricing, description and preview image.
- **Download ZIP File:** If the model is free, click the Download button. The ZIP file will be saved to your system.

8.3 Troubleshooting & Support

- **Can't Login?** Make sure you entered the correct email and password and try refreshing the page or clearing browser cache.
- **File Not Downloading?** Check your internet connection and make sure the model is free and available for download.
- **Need More Help?** Contact technical support via email use the Contact Us page on the website to submit a query.

1 User Register Page

3D Model Marketplace

Search "Architect"

[Login](#) [My Cart](#)

Create an Account

Name:
Enter your name

Email:
Enter your email

Password:
Enter your password

Confirm Password:
Enter confirm password

[Register](#)

Already have an account? [Login](#)

2 User Login Page

3D Model Marketplace

Search "Nature"

[Login](#) [My Cart](#)

Welcome to 3D Model Marketplace

You must be logged in to access this page

Email:
dhnul23@gmail.com

Password:
..

[Forgot password?](#)

[Login](#)

Don't have an account? [Register](#)

3 Dashboard

3D Model Marketplace

Search "Furniture"

[Account](#) [1 Item £142.00](#)

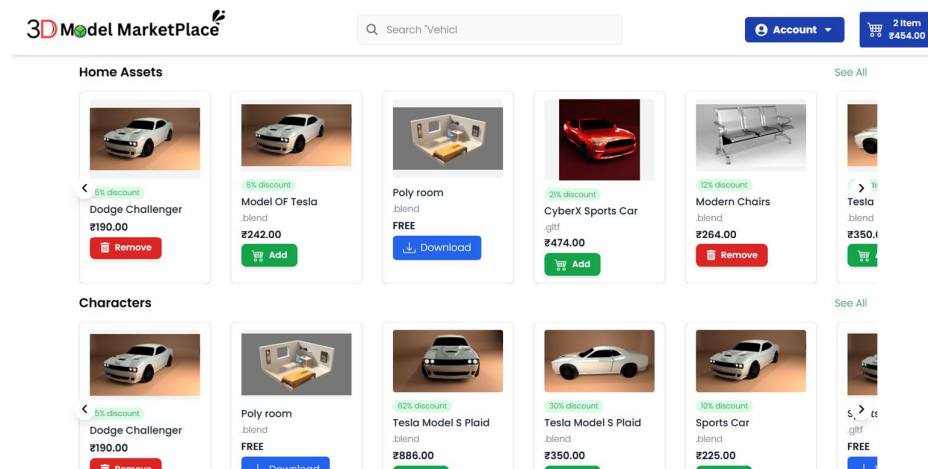
3D Models Provider Platform

BUY OR SELL YOUR MODELS THROUGH US

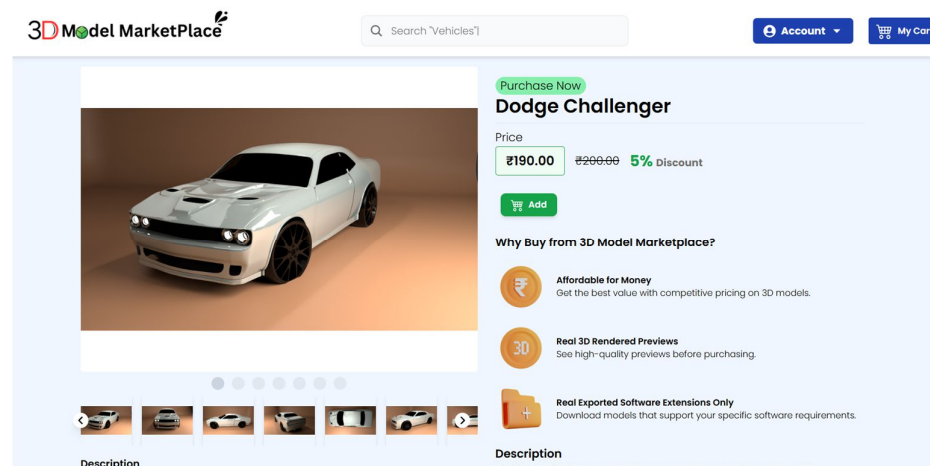
Vehicles [See All](#)

- [Dodge Challenger](#) 8% discount
- [Bugatti Chiron](#) 5% discount
- [Tesla Model S Plaid](#) 10% discount
- [Tesla Model S Plaid](#) 10% discount
- [Porsche 911 GT3 RS](#) 30% discount

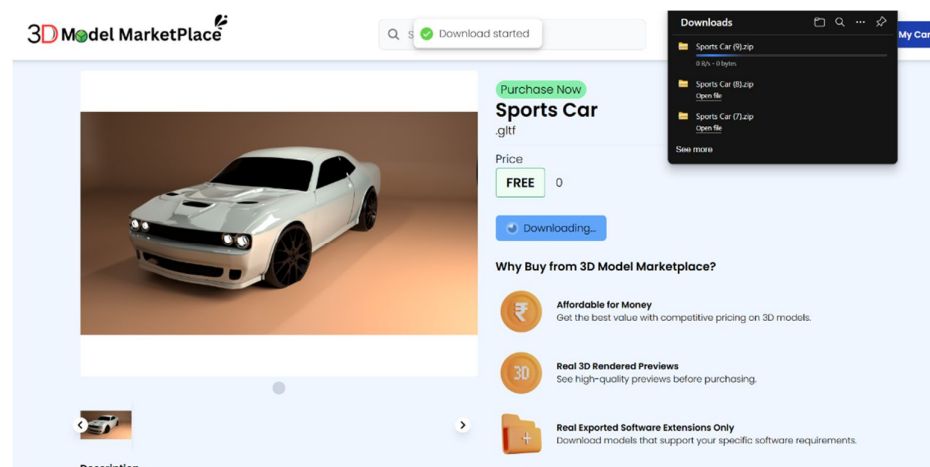
4 Category Wise Models Listed



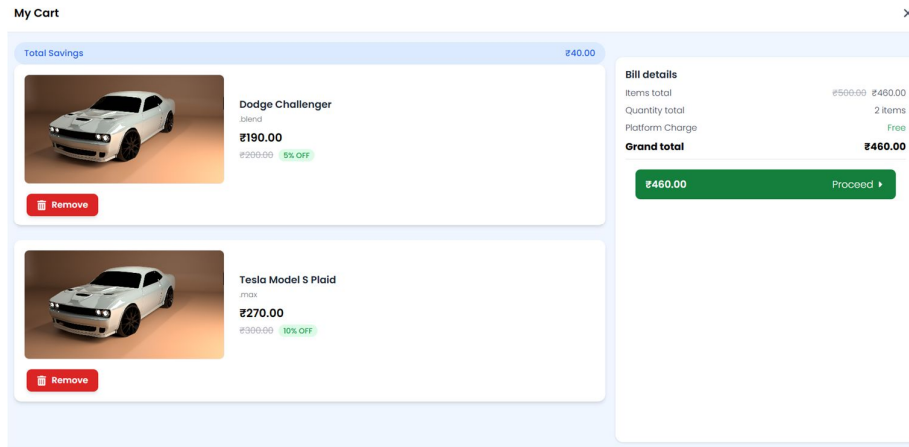
5 Model Details Page



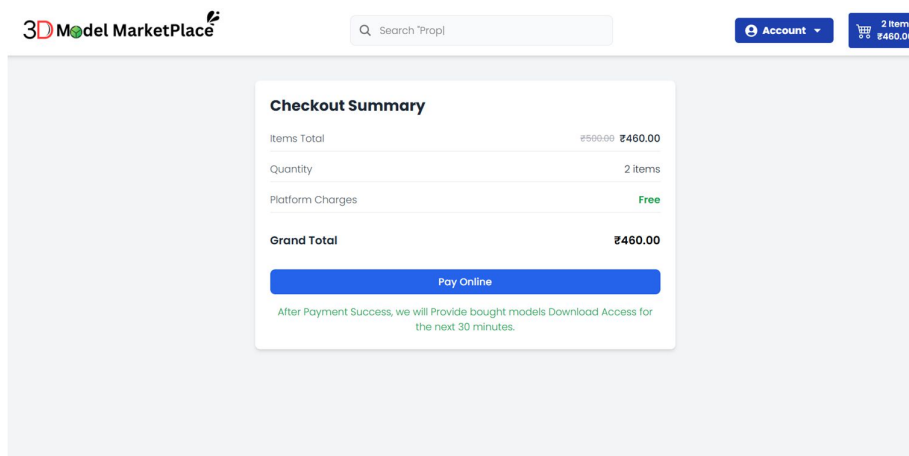
6 Directly Download Free Models



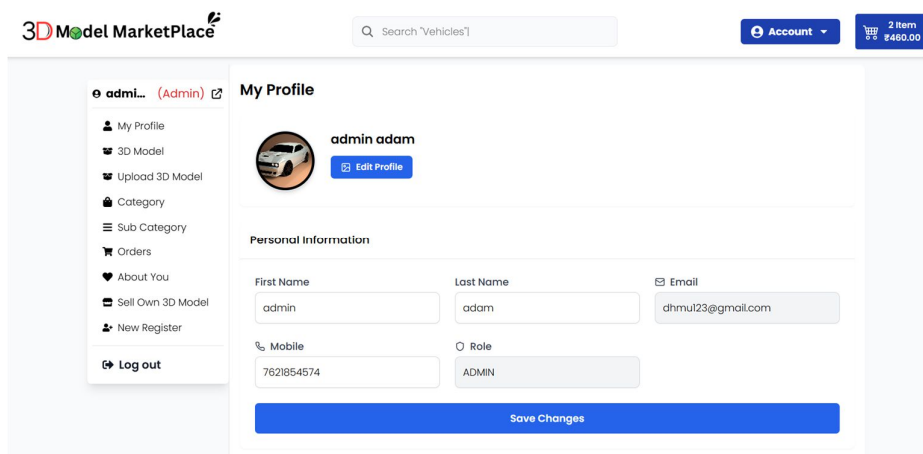
7 Cart for Paid Models



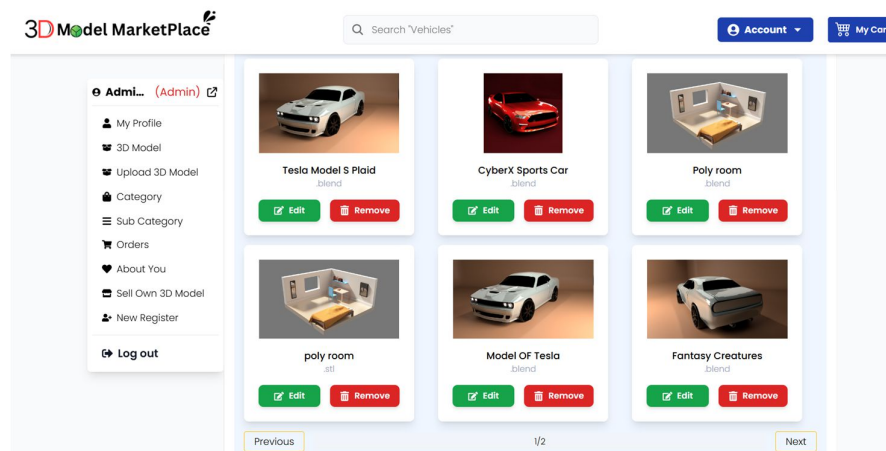
8 Checkout Page



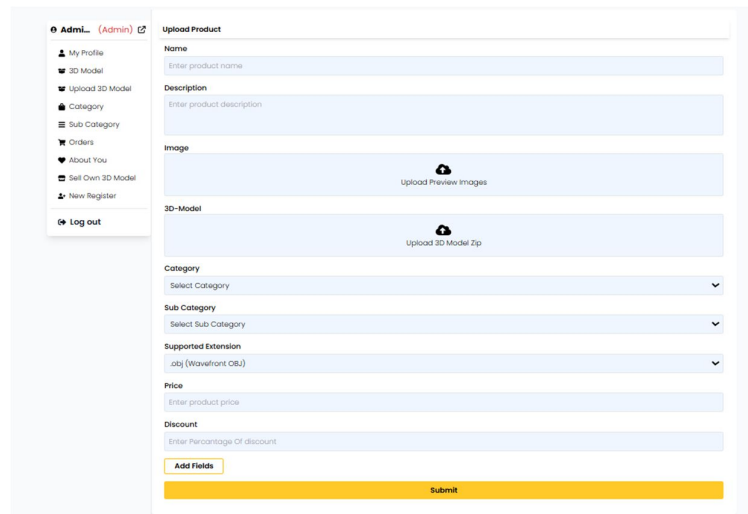
9 Admin Panel



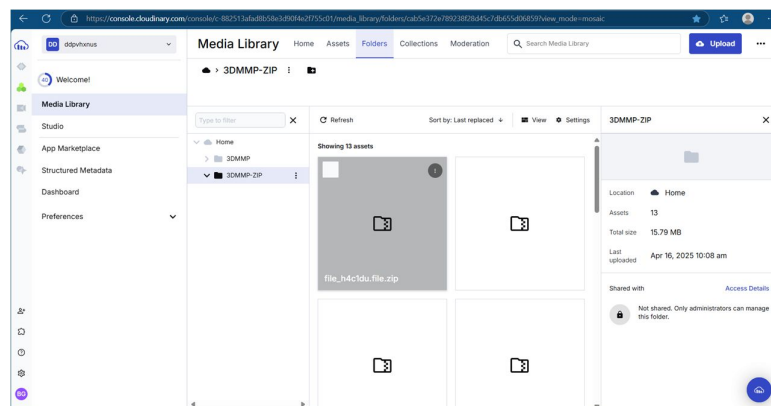
10 3D Models Manage Page



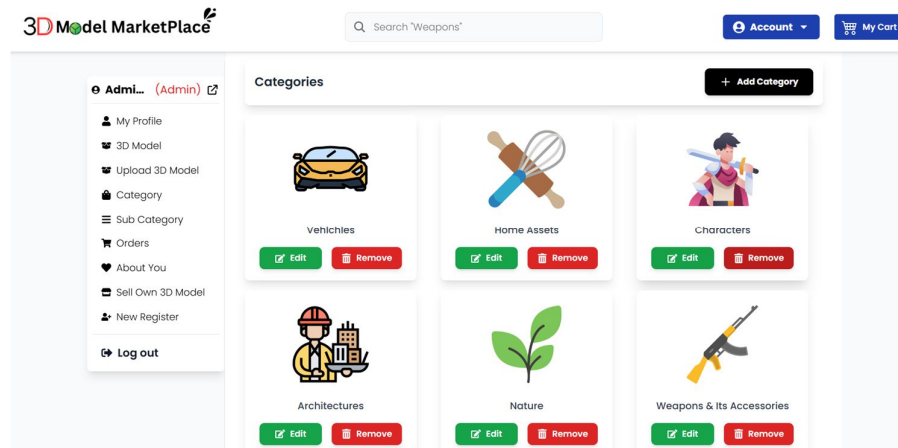
11 Upload 3D Model Page



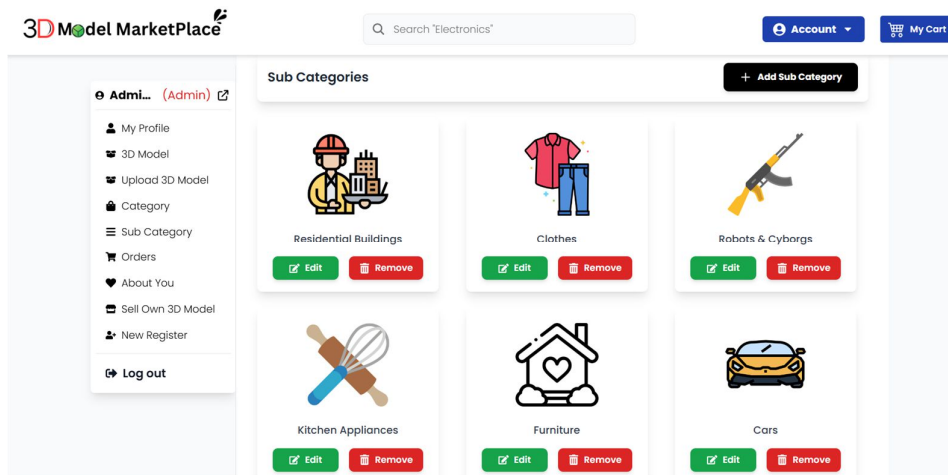
12 Cloudinary Storage for Zip Files



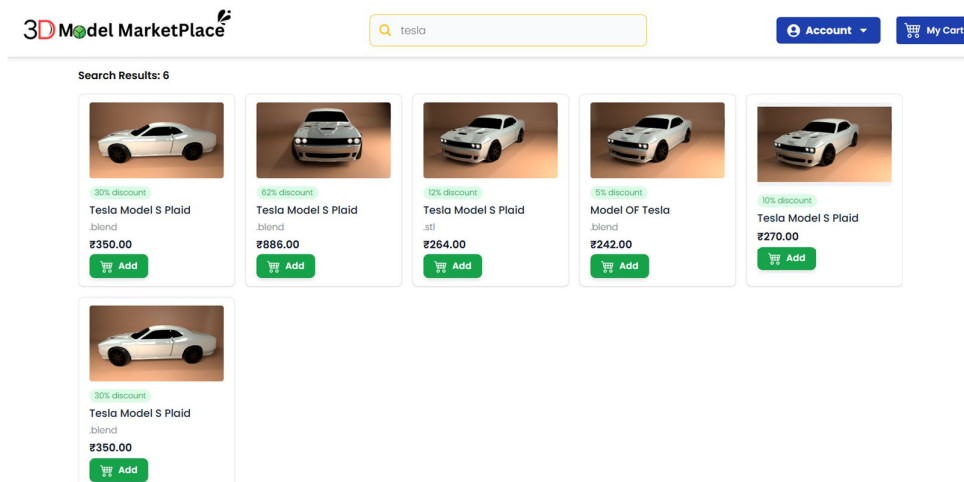
13 Category Management Page



14 Sub Category Management Page



15 Search Page



CHAPTER 9: LIMITATIONS AND FUTURE ENHANCEMENTS

9.1 Limitations

- **Internet Dependency:** The system requires a stable internet connection.
- **Limited Extension:** Currently sharing only one original software exported extension only.
- **File Format:** Limited single file format support (only ZIP files).
- **Transaction:** No payment gateway integration for paid models.
- **3D View:** Currently, only real rendered preview images should be shown, no 3D view should be provided.

9.2 Future Enhancements

- **Payment:** Integrate a payment gateway for paid models using platforms like Razorpay or PayPal.
- **Adding New Technologies:** Implement real-time 3D model rendering using **Three.js** JavaScript library that simplifies **WebGL** (web graphics library) for rendering high-performance 2D and 3D graphics in the browser.
- **Interactive 3D View:** This features interactive orbit controls for zooming, rotating, and panning within the 3D scene. It includes dynamic lighting, realistic shadows, and material effects, along with detailed textures and environment mapping. Animation playback is supported, and the scene is fully set up with camera and renderer configurations. The canvas is responsive, ensuring smooth rendering across modern browsers, which is attract to users to buy models.

CHAPTER 10: CONCLUSION AND DISCUSSION

10.1 Conclusion

This project presents a web-based platform where users can upload and download 3D model ZIP files. It offers category-wise and subcategory-wise browsing, user authentication, and admin-level file management. Cloudinary is used for storage, and the MERN stack ensures a robust structure.

10.2 Discussion

10.2.1 Self-Analysis of Project Viability

- **Managing dynamic categories and subcategories:** Implemented reusable schema and routes.
- **Download functionality security:** Ensured download access permission only for valid users.
- **Handling file size:** Added checks and compression mechanisms before upload.
- **Efficient Performance of data fetch:** Optimized backend queries and added pagination.
- **Access control:** Implemented user/admin middleware to restrict access properly.

10.2.2 Problems Encountered and Possible Solutions

Problem	Solution
Difficulty in integrating ZIP files	Solved by modifying upload logic and using Cloudinary API.
UI Responsiveness Issues	Used Tailwind CSS for better mobile compatibility.
Authentication Security	Implemented JWT-based authentication for secure access.

10.2.3 Summary of Project Work

- **Designed a full-stack MERN application** for uploading, managing, and downloading 3D model ZIP files.
- **Implemented Strong File Storage integration** for secure and scalable file storage.
- **Built a role-based access system** using JWT for secure user/admin operations.
- **Developed dynamic category & subcategory features** with reusable MongoDB schema and APIs.
- **Handled file upload logic and validations** to support only ZIP files with size restrictions.
- **Created responsive frontend** using React and Tailwind CSS for seamless UX across devices.
- **Ensured secure file downloads** by validating user sessions and model access conditions.
- **Optimized backend performance** with pagination and efficient MongoDB queries.
- **Maintained modular codebase** with clearly separated frontend, backend, and API layers.
- **Planned for scalability and enhancements**, including preview, reviews, and support for multiple languages.

REFERENCES

1. Tailwind CSS Features – <https://tailwindcss.com/>
2. Cloudinary API Features - <https://cloudinary.com/>