فصلنامه علمی (مقاله علمی-پژوهشی) جامعه‌شناسی سیاسی ایران، سال سوم،شماره سوم(پیاپی ۱۱)پاییز ۱۳۹۹،صص ۱۶۵۲-۱۶۳۳

طراحی و پیاده سازی و توسعه نرم افزار، تصور سازی داده ها با بهره گیری از شبیه سازی مجازی جهت نمایش اطلاعات آماری شهری

محسن فارسی[۱]

مهرداد متانی[۲]

علی فلاح[۳]

محمد رضا باقرزاده[۴]

یوسف قلی پور کنعانی[۵]

تاریخ دریافت :۲۳ /۱۳۹۹/۷

تاریخ پذیرش: ۱۳۹۹/۹/۴

چکیده

این مطالعه، طراحی، پیاده‌سازی و توسعه یک نرم‌افزار برای تصورسازی داده‌ها با استفاده از شبیه‌سازی مجازی به منظور نمایش اطلاعات آماری شهری است. در این پژوهش، یک ابزار توسعه داده شده است که به پژوهشگران امکان می‌دهد اطلاعات جمع‌آوری شده از منابع مختلف را با استفاده از شبیه‌سازی واقعیت مجازی مشاهده نمایند. با استفاده از این نرم‌افزار و قابلیت‌های آن، امکان تبدیل اعداد و داده‌های خام به مدل‌های سه‌بعدی در فضای واقعیت مجازی فراهم می‌شود. این امر باعث سهولت در درک رویدادهای به وقوع پیوسته در طول زمان می‌شود؛ زیرا بررسی، مشاهده و تحلیل اطلاعات خام مانند اعداد و جداول کاری دشوار است. با استفاده از این نرم‌افزار، پژوهشگران قادر خواهند بود رویدادها را به راحتی مشاهده نمایند و توانایی تحلیل را به طور قابل توجهی ارتقا دهند. موارد و روش ها: این ابزار از دو بخش اصلی تشکیل شده است که شامل: بخش ورود اطلاعات آنلاین و تحت وب: این بخش شامل یک پنل ورود اطلاعات به صورت آنلاین و تحت وب است. با استفاده از این بخش، کاربران قادرند به راحتی و با سهولت از ابزار استفاده کرده و اطلاعات خود را وارد کنند یا بارگذاری کنند. سیستم نمایش اطلاعات به صورت واقعیت مجازی: این بخش شامل یک سیستم است که اطلاعات را به صورت واقعیت مجازی نمایش می‌دهد. پژوهشگران می‌توانند اطلاعاتی که در این پلتفرم ثبت کرده‌اند را به صورت سه بعدی در نرم‌افزار شبیه‌سازی مجازی مشاهده کرده و به تحلیل وقایع ثبت شده بپردازند. پژوهشگران با استفاده از این دو بخش، قادرند اطلاعات خود را به صورت سه بعدی در محیط شبیه‌سازی مجازی ثبت و مشاهده کنند. از طریق تحلیل مشاهدات خود، می‌توانند اطلاعات و نتایج مورد نیاز خود را به دست آورده و به منظور درک بهتر برای دیگران، آنها را نمایش دهند.

کلید واژه: مدل سازی، شبیه سازی مجازی، تصور سازی، به کارگیری اطلاعات

[۱]دانشجوی دکتری، مدیریت رسانه ای، واحد قائم شهر، دانشگاه آزاد اسلامی، قائم شهر، ایران mohsen.farsi@gmail.com

[۲]استادیار گروه مدیریت رسانه، واحد قائم شهر، دانشگاه آزاد اسلامی، قائم شهر، ایران (نویسنده مسئول) mehrdadmatani@yahoo.com

[۳]استادیار گروه مدیریت رسانه، واحد نور شهر، دانشگاه آزاد اسلامی، نور، ایران falahali55@yahoo.com

[۴]استادیار گروه مدیریت دولتی، واحد قائم شهر، دانشگاه آزاد اسلامی، قائم شهر، ایران thinks.paper@yahoo.com

[۵]استادیار گروه مهندسی صنایع، واحد قائم شهر، دانشگاه آزاد اسلامی، قائم شهر، ایران golipourkanani@yahoo.com

۱- مقدمه:

استفاده از ابزار ها، نرم افزارها[1] و پلتفرمهای[2] جدید به منظور تسهیل فرآیند کار و افزایش سرعت آن، امروزه بسیار حائز اهمیت است. هر روز، کاربران و پژوهشگران سعی می‌کنند بیشترین استفاده را از این ابزارها و پلتفرم‌ها بهره‌برداری کنند تا در کمترین زمان ممکن به بهترین نتیجه برسند.

طراحی پلتفرم‌ها و نرم افزارها به گونه‌ای است که توانمندی‌هایی را برای کاربران و پژوهشگران ایجاد نماید، به نحوی که با استفاده از آن‌ها بتوانند به نتایج مناسب و تحلیل‌های مطلوب دست یابند.

استفاده از نرم افزارها و ابزارهای تحلیل[3] و نمایش داده به پژوهشگران کمک می‌کند تا با سهولت به تحلیل اعدادی که از منابع مختلف دریافت می‌کنند، بپردازند. با این حال، هر نرم افزار و ابزاری قابلیت‌ها و محدودیت‌های خاص خود را دارد. استفاده از تکنولوژی‌های جدید مانند شبیه سازی مجازی[4] می‌تواند در پاسخگویی به برخی از نیازهای پژوهشگران در نمایش اطلاعات پیچیده کمک کرده و نتایج و رویدادها را به نحوی بهتر و قابل فهم‌تری برای آن‌ها به نمایش بگذارد.

در این پژوهش، هدف ما ارائه و ایجاد یک پلتفرم است که به کاربران و پژوهشگران امکان می‌دهد بدون نیاز به دانش فراوان و بدون درگیری با پیچیدگی‌های ریاضی و برنامه‌نویسی[5]، به سادگی داده‌های خام خود را به نحوی قابل فهم و آمیزش برای خود و دیگران نمایش دهند و به راحتی و تحلیل اطلاعات وارد شده در فضای واقعیت مجازی بپردازند.

در این مقاله، ابتدا به بیان مساله اصلی این پژوهش می‌پردازیم و سپس به اهمیت و ضرورت طراحی این پلتفرم می‌پردازیم. همچنین، تعاریفی از مفاهیم مورد استفاده در این مقاله را بررسی خواهیم کرد تا درک بهتری از جنبه‌های مختلف این پژوهش به دست آوریم.

۲- موارد و روش ها:

امروزه، منابع مختلفی از اطلاعات، از جمله سنسورها[6]، اطلاعات ذخیره شده و پردازش شده، و همچنین اطلاعات خام[7] دریافت شده از منابع مختلف، قابل دریافت و ذخیره‌سازی هستند. این اطلاعات اغلب به صورت داده‌های متنی[8] ذخیره می‌شوند که به راحتی قابل درک و پردازش برای انسان‌ها و پژوهشگران نیست. بنابراین، پژوهشگران تلاش می‌کنند تا این اطلاعات خام متنی را به شکل‌های مختلفی مانند نمودارها و تصاویر تبدیل کنند تا با استفاده از این نمودارها و تصاویر، به درک بهتری از اطلاعات دست یابند. با این حال، همیشه نمودارها و تصاویر کافی نیستند و برای درک کامل اطلاعات اغلب ناکافی هستند. به علت تغییرات زمانی و رخدادهای همزمان در برخی موارد، و همچنین وجود اطلاعات در فضای سه بعدی، نمودارها قادر به نمایش کامل این اطلاعات نیستند. هدف این پژوهش ایجاد یک پلتفرم است که به پژوهشگران امکان می‌دهد تا این اطلاعات را به صورت سه بعدی در فضای واقعیت

---

[1] Software
[2] Platforms
[3] Analysis
[4] Visualization Simulation
[5] Programming
[6] Sensor Data
[7] Raw Data
[8] Text Data

مجازی مشـــاهده کنند و به راحتی به تحلیل و مشـــاهده تمامی رخدادها در یک لحظه بپردازند.

برای طراحی این نرم‌افزار، به دو بخش متفاوت نیاز داریم:

بخش اول: پنل ورود اطلاعات به صورت آنلاین و تحت وب است. این بخش برای کاربران راحتی کار و سهولت استفاده از این ابزار را فراهم می‌کند. کاربران می‌توانند به راحتی اطلاعات خود را اضـــافه کرده و یا آن‌ها را بارگذاری کنند.

بخش دوم: سیستم نمایش اطلاعات به صورت واقعیت مجازی است. این سیستم به پژوهشگران امکان می‌دهد اطلاعاتی که در این پلتفرم ثبت کرده‌اند را به صورت سه بعدی در یک نرم‌افزار شبیه‌سازی مجازی مشاهده کنند و به تحلیل وقایع ثبت شده بپردازند.

تعاریف واژه ها و دلایل های استفاده:

مدل ســـازی: معنای لغت "مدل" از ریشـــه لاتینی "Modus" به معنای "اندازه گرفته شـــده" گرفته شـــده اســـت. همچنین، مدل به ما کمک می‌کند تا به متن و به درون پدیده‌هایی که نمی‌توانیم به طور مســـتقیم آن‌ها را ببینیم، هدایت شویم. مدل، در واقع یک نسخه کوچک یا بازســازی کوچکی از یک شــیء بزرگ اســت که از نظر عملکرد با شــیء واقعی یکســان اســت.[1]. در این متن، روابط اصـــلی که در جنبه‌های مورد بحث بیان شـــده را روشـــن می‌کند و در نهایت، امکان آزمایش تجربی تئوری را با توجه به ماهیت این روابط فراهم می‌کند. پس از آزمایش مدل، درک بهتری از برخی بخش‌های واقعی دنیا به دســت می‌آید. به طور خلاصــه، می‌توان گفت که مدل یک سیستم است که شامل مفاهیم، فرضیه‌ها و شاخص‌ها است و به کمک انتخاب و جمع‌آوری اطلاعات مورد نیاز برای آزمون فرضـــیه‌ها کمک می‌کند.[2]. مدل‌ها به طور دقیق و کامل از واقعیات و روابط اصـــلی بین اجزای یک پدیده پیروی می‌کنند و نمای کلی از واقعیت را به ما ارائه می‌دهند. .]3[مدل، یک نسخه کوچکتر یا بازسازی شده از یک شـیء یا پدیده بزرگ است که در عملکرد خود با آن شـــیء یا پدیده واقعی همانندی دارد. این به ما کمک می‌کند که در شـرایطی که دسـترسـی به تمام جزئیات و روابط پدیده‌ها مشـکل، پرهزینه و زمان‌بر اسـت، مدل آن را آســـان و ســـاده می‌کند و با داشـــتن قابلیت تحلیل و تجزیه، پیش‌بینی نتایج را ممکن می‌سـازد.]4[. مدل، مانند یک اسکلت برای نظریه استفاده می‌شود. یک مدل قابل اعتماد باید بتولند در پیش‌بینی رویدادها کمک کند و این صفت پیش‌بینی کننده ٔ مدل سه خصوصیت را شامل می‌شود: دقت بالا، تنوع غنی در ترکیبات و حداکثر سطح تناسب و یا قدرت سازمان‌دهی.]5[. مدل‌های گرافیکی به عنوان جایگزینی برای واقعیت، تصـــویر را به نمایش می‌گذارند. این مدل‌ها، رابطه بین متغیرها و پارامترها را در قالب تصـــاویری مانند عکس‌ها و نقاشـــی‌ها نمایش می‌دهند..]6[

مدل‌ها قابل تقسـیم به دو دسـته، یعنی مدل‌های فیزیکی (ماننـد مـدل‌هـای هواپیمـا) و مـدل‌هـای انتزاعی (ماننـد بیانیه‌های ریاضی که الگوهای رفتاری را توصیف می‌کنند) هســـتند. مدل‌های انتزاعی یا مفهومی، نقش مهمی در فلسـفه علم دارند. به طور خلاصـه، می‌توان مدل‌ها را به دو دسته تقسیم کرد: مدل‌های فیزیکی که شامل اشیاء فیزیکی مانند مدل‌های هواپیما هستند، و مدل‌های انتزاعی که شامل بیانیه‌ها یا عبارات ریاضی هستند که الگوهای رفتاری را توصیف می‌کنند.][۷][۸].

دلایل استفاده از مدل سازی: ۱- دید واضح و شفاف به ســـیســـتم ارائه می‌دهد که باعث حذف پیچیدگی‌ها و نقاط تاریک سـیسـتم می‌شـود. ۲- معماری و چارچوب سیستم را مشخص می‌کند و این امکان را فراهم می‌کند

که زمان و نیروی انسانی مورد نیاز برای پروژه را تخمین بزنیم. ۳- امکان مدیریت ریسک وجود دارد. با بررسی موارد ۱ و ۲ می‌توانیم ریسک‌های سیستم را شناسایی و مدیریت کنیم. تعریف ریسک: هر چیزی که ممکن است مشکلات زمان و بودجه پروژه را ایجاد کند را ریسک می‌نامیم. ریسک شناخته شده به عنوان معروف‌ترین ریسک مطرح است. این ریسک زمانی رخ می‌دهد که نیازهای کاربر (به هر دلیلی) به طرز نادرستی در نظر گرفته شوند. ۴- ایجاد مستندات و دستیابی به قابلیت استفاده مجدد (Reuse) موجب می‌شود که مدل‌سازی شما به سمت ایجاد مستندات هدایت شود. این نتایج همه به سمت رسیدن به قابلیت استفاده مجدد معطوف می‌شوند و از تکرار کارها جلوگیری می‌کنند.

شبیه سازی[9]: مفهوم کلمه "Simulation" به معنای "بیان واقعیتی بدون وجود واقعیت" است.]9[مفهومی که عبارت "نمایش کارکرد یک سیستم یا یک فرایند با استفاده از کامپیوتر یا روش دیگر" را توصیف می‌کند، به معنای نمی‌توانستن اصل یک سیستم یا فرایند را به صورت واقعی تجربه کنیم ولی با استفاده از کامپیوتر یا روش‌های دیگر، عملکرد آن را تجسم می‌کنیم.]10[بیان شده است که شبیه‌سازی قابلیت و توانایی طراحی را دارد؛ این به معنای ایجاد یک راه‌حل آماری قدرتمند است که مدیر را قادر می‌سازد به دست‌یابی به اهداف سازمان اطمینان حاصل کند..]11[یک مدل شبیه‌سازی به طور کلی نمایشی از پدیده‌ها یا سیستم‌های دینامیک را ارائه می‌دهد. این نوع مدل قادر است مسائل موجود در سازمان را قبل از تبدیل شدن به مشکل، شناسایی کند. شبیه‌سازی در واقع فرآیند طراحی مدلی از سیستم واقعی است. با استفاده از این مدل و انجام آزمایش‌ها، هدف آن پی‌بردن به رفتار سیستم یا ارزیابی استراتژی‌های مختلف در محدوده‌ای است که توسط معیار یا مجموعه‌ای از معیارها تعریف شده است. این فرآیند برای عملیات سیستم به کار می‌رود و به منظور بررسی و ارزیابی صحت عملکرد سیستم مورد استفاده قرار می‌گیرد. [12]

شبیه‌سازی تقلیدی، به معنای تقلید و شبیه‌سازی عملکرد یک فرآیند یا سیستم واقعی با گذشت زمان است. در این روش، رفتار و عملکرد سیستم با استفاده از ساختن یک مدل شبیه‌سازی بررسی می‌شود. این مدل معمولاً شامل مجموعه‌ای از فرضیات مرتبط با عملکرد سیستم است که به صورت رابطه‌های ریاضی، منطقی و نمادین بین نهاده‌ها و هدف‌های مورد نظر سیستم بیان می‌شود. این فرضیات به عنوان اصول و قوانینی که سیستم بر اساس آنها عمل می‌کند، مورد استفاده قرار می‌گیرند. از طریق شبیه‌سازی تقلیدی، ما می‌توانیم رفتار و عملکرد سیستم را در طول زمان به دقت بررسی کرده و نتایجی را که در شرایط واقعی ممکن است دشوار یا پرهزینه باشد، به دست آوریم..[13]

دلایل استفاده از شبیه سازی: شبیه‌سازی یک تکنیک کمّی است که برای مطالعه و ارزیابی گزینه‌های مختلف استفاده می‌شود. این روش، از طریق ساخت و اجرای مدل‌های شبیه‌سازی بر روی سیستم واقعی، به منظور پیش‌بینی رفتار آینده سیستم، امکان‌پذیر می‌شود. به طور خلاصه، دلایل استفاده از شبیه‌سازی عبارتند از:

۱. عدم امکان استفاده از مدل‌های تحلیلی برای ارزیابی و بهبود فرآیندهای تولید پایگاه‌های اطلاعاتی.

۲. امکان تغییر ساختار مدل به راحتی در شبیه‌سازی، که به سوالات مختلف درباره عواقب احتمالی تغییرات در سیستم واقعی پاسخ داده می‌شود. این خصوصیت

[9] Simulation

شـبیه‌سـازی در مواجهه با محیط‌های پویا و پراشـتباه با متغیرهای متغیر بسیار مؤثر است.

۳. اسـتفاده از شـبیه‌سـازی در مواردی که هزینه اعمال تغییرات پیشنهادی زیاد است، مفید است. در برخی موارد که سـیسـتم هنوز در عمل واقعی خلق نشـده و فقط اطلاعاتی درباره روابط نظری آن وجود دارد، شـبیه‌سـازی تنها راه‌حل است.

۴. در شبیه‌سازی، امکان فشرده‌سازی زمان وجود دارد.

۵. توصیف مدل‌های پیچیده ریاضی فعالیت‌های مرتبط با مدل‌سازی ریاضی، زمان‌بر و پیچیده است. اما توصیف مدل شـبیه‌سـازی فعالیت‌های مرتبط با مدل‌سـازی به سادگی امکان‌پذیر است و نیاز به زمان کمتری دارد. [14]

همچنین، می‌توانیم بیافزاییم که برای دسـتیابی به موفقیت یک پروژه شـبیه‌سـازی، لازم اسـت مدل ایجاد شـده قابل پذیرش و قابل فهم باشـد و در عمل اسـتفاده شـود. مسـتندسـازی دقیق و جامع درباره روش ایجاد، توسـعه و عملکرد مدل می‌توانـد عمر مفیـد و احتمـال موفقیـت در پیاده‌سازی آن را افزایش دهد.[15]

در شـبیه‌سـازی کامپیوتری، مدل سـاخته شـده یک برنامه کامپیوتری است که تمام اجزا و ویژگی‌های سیستم را به سـاختارهای برنامه‌نویسـی و توابع ریاضـی تبدیل می‌کند. در داخل این برنامه، قوانین و روابط حاکم بر سـیسـتم و ارتباطات بین آنها در نظر گرفته می‌شـوند. شـبیه‌سـازی کامپیوتری به دلیل کاربرد عملی و ویژگی‌های منحصـر به فرد خود، برای بررسـی و مطالعه انواع سیسـتم‌ها اسـتفاده می‌شـود، از جمله حمل و نقل، بیمارسـتان، سـیسـتم‌های صنعتی، تولیدی، ترافیک، انبار و غیره. [16]

تصور سازی[10]: تصورسازی اطلاعات به معنای ارائه داده‌ها در قالب تصـویری و گرافیکی اسـت. این روش ارائه به تصمیم‌گیران و صاحبان اطلاعات امکان می‌دهد تا تحلیل داده‌های خود را به صـورت بصـری دریافت کنند. این امر باعث می‌شـود تا فهم مفاهیم پیچیده و شـناسـایی الگوهای جدید برای آنها آسان‌تر شـود. اثبات شـده اسـت که ما با تماشـای تصـاویر بهتر می‌توانیم اطلاعات را درک کنیم. تماشـای یک تابلو و درک آن سـریعتر و سـاده‌تر از خواندن یک کتاب در ذهن ما تأثیر می‌گذارد. از این رو، با اسـتفاده از تجسـم تعاملی داده‌ها، همواره به سـمت جلو حرکت می‌کنیم.

در ابتدا، گرافیک رایانه‌ای برای حل مشـکلات علمی استفاده می‌شد. با این حال، در آغازین روزهای استفاده از گرافیک رایانه‌ای، محدودیت قدرت گرافیکی آن بهره‌وری آن را محدود می‌کرد. تأکید اخیر بر مصـورسـازی در سـال ۱۹۸۷ با انتشـار مقاله "مصـورسـازی در محاسـبات علمی" به عنوان یک ویژه‌نامه مربوط به گرافیک رایانه‌ای آغاز شد.]17[

مصـورسـازی علمی به عنوان یک موضـوع در علوم کامپیوتر، اسـتفاده از بازنمودهای حسـی تعاملی، به طور معمول بصـری، از داده‌های انتزاعی برای تقویت شـناخت، فرضـیه‌سـازی و اسـتدلال اسـتفاده می‌کند. مصـورسـازی داده‌ها یک زیرمجموعه از مصورسازی است که با استفاده از گرافیک آماری و داده‌های جغرافیایی یا مکانی (مانند نقشـه‌های مفهومی)، به طور خلاصـه و انتزاعی ارائه می‌شود.]۱۸[

مصورسازی علمی یک فرآیند است که به وسیله آن، داده‌های حاصـل از شـبیه‌سـازی‌ها یا آزمایش‌ها به وسـیله یک سـاختار هندسـی، ضـمنی یا صـریح، تصـویرسـازی

[10] Visualization

می‌شـــوند تا بتوان آنها را کاوش، تجزیه و تحلیل کرده و درک بهتری از آنها پیدا کرد. مصورسازی علمی بر تبدیل داده‌های پیچیده به روش‌های گرافیکی و انیمیشـن تمرکز دارد تا بتوان به صـــورت قابل فهم و شـــفافی داده‌های با ابعاد بالاتر را نشان داد. [20] [19]

بنابراین، مصورسازی که به درستی طراحی می‌شود، نه تنها به ارائه یک بخش مهم از تجزیه و تحلیل داده‌ها کمک می‌کند، بلکه در عین حال فرآیند انتقال دانش را نیز انجام می‌دهد. [21]

برای مصورسازی یک میدان اسکالر سه بعدی، می‌توان از روش‌هـای مختلفی اســـتفـاده کرد. یـک روش متـداول، استفاده از سطح‌های ایزو برای نمایش توزیع میدان است. در این روش، ســطوحی که مقدار میدان اســکالر در آن‌ها یکسان است، به عنوان سطح‌های ایزو انتخاب می‌شوند و از طریق آنها توزیع میدان را نشـــان می‌دهیم. همچنین، می‌توان از بافت‌ها برای نمایش شیب میدان استفاده کرد. بافت‌ها، نواحیی هستند که تغییرات شدت میدان را نشان می‌دهند و با اســـتفاده از آنها، می‌توان شـــیب میدان را در نقـاط مختلف نمـایش داد. این روش‌هـا می‌تواننـد در مصـورسـازی میدان‌های اسـکالر سـه بعدی مفید باشـند. [22]اسـتفاده از قالب‌های تصـویری متنوعی می‌تواند در مصـــورســـازی اطلاعات به کار گرفته شـــود. این قالب‌ها شامل طرح‌ها، نمودارها، تصاویر، اشیا، تجسم‌های تعاملی، برنامه‌های تجســـم اطلاعات و تجســـم‌های خیالی مانند داستان‌ها می‌شـوند. هدف اصـلی مصـورسـازی اطلاعات، اسـتفاده از ابزارهای پشـتیبانی شـده رایانه‌ای اسـت تا به دست آوردن دید جدیدی را ممکن سازد. تجزیه و تحلیل بصری به تعامل انسان با سیستم‌های تجسم متمرکز است و به عنوان بخشی از فرایند بزرگتر تجزیه و تحلیل داده‌ها در نظر گرفته می‌شـود. تجزیه و تحلیل بصـری به عنوان "علم استدلال تحلیلی پشتیبانی شده توسط رابط بصری تعاملی" تعریف شـده اسـت. این تعریف نشـان می‌دهد که تجزیه و تحلیل بصـری، از طریق تعامل بصـری و اسـتفاده از رابط‌هـای تعـاملی، بـه ارائـه اســـتـدلال‌هـا و تحلیل‌هـای تصویری کمک می‌کند. [23]

دلایل استفاده از تصور سازی:**1**

۱. تسهیل در ارائه اطلاعات
۲. سرعت بالا در ارائه اطلاعات
۳. جذابیت و جالب بودن برای کاربران
۴. کاهش خستگی در ارتباط با اطلاعات
۵. ارائه اطلاعات فراوان در زمان محدود
۶. قابلیت نمایش ســـریع، روشـــن و زیبا برای ایده‌های پیچیده
۷. امکان برقراری آسـان ارتباط بین عناصـر ارائه شـده در موضوع
۸. استفاده از خلاقیت در تصمیم‌گیری

طراحی و پیاده سازی نرم افزار:

با توجه به نکات مطرح شـــده، طراحی و پیاده‌ســـازی نرم‌افزار به شرح زیر است:

در این مقـالـه، برای بخش ورود اطلاعـات از فریم‌ورک دانت کور مایکروسافت و برای بخش نمایش اطلاعات به صـورت واقعیت مجازی از موتور یونیتی اسـتفاده خواهیم کرد.

بخش اول: طراحی وب پنل

در این بخش، مـا مـدل‌هـای مختلفی را تعریف خواهیم کرد. اشیا به صـورت مدل‌های هندسـی سـاده و پیچیده در فضای پیرامونی ما وجود دارند. برخی از اجسام سه‌بعدی سـاده عبارتند از: مکعب، کُره، کپسـول، اسـتوانه، صـفحه و غیره.

برای تعریف این اجسام در پنل دیتا، نیاز به پیاده‌سازی یک مدل داریم. این مدل شامل ویژگی‌های زیر است:

۱- شناسه یکتا برای هر شیء: این فیلد به صورت یکتا ایجاد می‌شود و به هر جسم یک کد منحصربه‌فرد تخصیص داده می‌شود.

۲- نام شیء: این فیلد برای تعریف عنوان هر جسم استفاده می‌شود.

۳- تصویر شیء: این فیلد برای آپلود تصویر هر جسم استفاده می‌شود تا جسم‌ها در پنل به راحتی شناسایی شوند.

۴- آدرس شیء: این فیلد برای اشیایی که در خارج سیستم ایجاد شده‌اند و در آینده به سیستم اضافه خواهند شد، در نظر گرفته شده است.

برای پیاده‌سازی این ویژگی‌ها، ابتدا باید یک مدل استاندارد تعریف کنیم که به نام "ObjectType" است و ویژگی‌های زیر را به این بخش اضافه می‌کنیم.

1-ObjectTypeID 2- Name 3- Picture 4- URL

```
using System.ComponentModel.DataAnnotations;

namespace VisualizationDataPanel.Models
{
    18 references
    public class ObjectType
    {
        [Key]
        11 references
        public int ObjectTypeID { get; set; }
        12 references
        public string Name { get; set; }
        12 references
        public string? Picture { get; set; }
        12 references
        public string? URL { get; set; }
    }
}
```

کد ۱ – کلاس مدل *ObjectType*

تعریف نمونه از اجسام: در این مقاله، به تعریف نمونه‌های از اجسام سه‌بعدی پرداخته می‌شود. تمامی اجسام سه‌بعدی ویژگی‌های مشترکی را دارند که آنها را در دسته‌بندی‌های زیر تقسیم می‌کنند:

۱- نوع شکل اجسام سه‌بعدی،

۲- مختصات اجسام در فضای سه‌بعدی،

۳- چرخش اجسام در فضای سه‌بعدی،

۴- اندازه اجسام در فضای سه‌بعدی،

۵- رنگ اجسام در فضای سه‌بعدی. با در نظر گرفتن این ویژگی‌ها، می‌توانیم اجسام مختلف را در فضای سه‌بعدی تعریف کرده و به نمایش بیاوریم.

برای تعریف نمونه‌های از اجسام در پنل داده، نیاز به پیاده‌سازی یک مدل داریم. این مدل شامل ویژگی‌های زیر است:

۱- شناسه یکتا برای هر نمونه اجسام: این فیلد به هر نمونه از اجسام تعریف شده در فضای سه‌بعدی یک کد یکتا اختصاص می‌دهد تا به راحتی قابل شناسایی باشد.

۲- نام نمونه جسم: این فیلد برای هر نمونه یک نام رشته‌ای تعریف می‌کند تا تشخیص و شناسایی آن در فضای سه‌بعدی برای کاربران آسان باشد.

۳- نوع اجسام: این فیلد از مدل‌های پیش‌فرض در سیستم استفاده می‌کند تا نیازی به تعریف مجدد هر آبجکت نباشد. این روش سرعت تعریف اجسام را افزایش می‌دهد و در صورت تغییر مدل‌های نمونه، تغییرات در تمام جاهایی که از آن مدل نمونه استفاده شده است، اعمال می‌شود.

۴- مختصات مکانی: این فیلد هر نمونه از مدل را به صورت پیش‌فرض در یک مختصات اولیه تعریف شده (شامل محورهای **x**،**y** و **z**) قرار می‌دهد.

۵- مختصات چرخشی: این فیلد هر نمونه از مدل را به صورت پیش‌فرض در یک زاویه چرخشی اولیه تعریف شده (نسبت به محورهای **x**،**y** و **z**) قرار می‌دهد.

۶- اندازه جسم: این فیلد هر نمونه از مدل را به صورت پیش‌فرض با استفاده از یک نسبت ابعادی نسبت به ابعاد متریک استاندارد تعریف می‌کند.

۷- رنگ جسم: این فیلد به هر نمونه از مدل را به صورت پیش‌فرض یک رنگ RGB اختصاص می‌دهد تا جسم با آن رنگ نمایش داده شود.

به منظور پیاده‌سازی این ویژگی‌ها، ابتدا باید یک مدل استاندارد تعریف کنیم. مدل را با نام "MyObject" مشخص می‌کنیم و ویژگی‌های زیر را به این مدل اضافه می‌کنیم:

1-MyObjectTypeID 2-Name 3-ObjectType 4-Position 5-Rotation 6-Scale 7-Color 8-Data

```
using System.ComponentModel.DataAnnotations;

namespace VisualizationDataPanel.Models
{
    public class MyObject
    {
        [Key]
        public int MyObjectID { get; set; }
        public string Name { get; set; }
        public ObjectType? objectType { get; set; }
        public int ObjectTypeID { get; set; }
        public float PostionX { get; set; }
        public float PostionY { get; set; }
        public float PostionZ { get; set; }
        public float RotationX { get; set; }
        public float RotationY { get; set; }
        public float RotationZ { get; set; }
        public float ScaleX { get; set; }
        public float ScaleY { get; set; }
        public float ScaleZ { get; set; }
        public byte R { get; set; }
        public byte G { get; set; }
        public byte B { get; set; }
        public byte O { get; set; }
        public string Data { get; set; }
```

کد ۲ – کلاس مدل *MyObject*

ایجاد جدول ها در پایگاه داده ها: **Database**

در این بخش، ما به تعریف و ایجاد جدول‌ها برای موجودیت‌ها در پایگاه داده Microsoft SQL Server می‌پردازیم. ابتدا جدول "ObjectType" را با استفاده از کدهای زیر در پایگاه داده ایجاد می‌کنیم:

```
USE [VisualizationDataPanelDB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ObjectType](
    [ObjectTypeID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [nvarchar](max) NOT NULL,
    [Picture] [nvarchar](max) NULL,
    [URL] [nvarchar](max) NULL,
 CONSTRAINT [PK_ObjectType] PRIMARY KEY CLUSTERED
(
    [ObjectTypeID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

کد ۳ – دستورات ایجاد جدول *ObjectType* در پایگاه داده

پس از اجرا این قطعه کد، جدول مرتبط با Object Type در پایگاه داده ها ایجاد میشود.

Column Name	Data Type	Allow Nulls
ObjectTypeID	int	☐
Name	nvarchar(MAX)	☐
Picture	nvarchar(MAX)	☑
URL	nvarchar(MAX)	☑
		☐

شکل ۱ – جدول ایجاد شده *ObjectType* در پایگاه داده

جهت تعریف جدول MyObject در این قسمت ما جدول MyObject را با استفاده از کد های زیر در پایگاه داده ایجاد میکنیم.

```
USE [VisualizationDataPanelDB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[MyObject](
    [MyObjectID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [nvarchar](max) NOT NULL,
    [ObjectTypeID] [int] NOT NULL,
    [PostionX] [real] NOT NULL,
    [PostionY] [real] NOT NULL,
    [PostionZ] [real] NOT NULL,
    [RotationX] [real] NOT NULL,
    [RotationY] [real] NOT NULL,
    [RotationZ] [real] NOT NULL,
    [ScaleX] [real] NOT NULL,
    [ScaleY] [real] NOT NULL,
    [ScaleZ] [real] NOT NULL,
    [R] [tinyint] NOT NULL,
    [G] [tinyint] NOT NULL,
    [B] [tinyint] NOT NULL,
    [O] [tinyint] NOT NULL,
    [Data] [nvarchar](max) NOT NULL,
 CONSTRAINT [PK_MyObject] PRIMARY KEY CLUSTERED
(
    [MyObjectID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOC
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
ALTER TABLE [dbo].[MyObject]  WITH CHECK ADD  CONSTRAINT [FK_MyObject_ObjectType_ObjectTy
REFERENCES [dbo].[ObjectType] ([ObjectTypeID])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[MyObject] CHECK CONSTRAINT [FK_MyObject_ObjectType_ObjectTypeID]
GO
```

کد ۴ – دستورات ایجاد جدول *MyObject* در پایگاه داده

پس از اجرا این قطعه کد، جدول مرتبط با MyObject در پایگاه داده ها ایجاد میشود.

Column Name	Data Type	Allow Nulls
MyObjectID	int	☐
Name	nvarchar(MAX)	☐
ObjectTypeID	int	☐
PostionX	real	☐
PostionY	real	☐
PostionZ	real	☐
RotationX	real	☐
RotationY	real	☐
RotationZ	real	☐
ScaleX	real	☐
ScaleY	real	☐
ScaleZ	real	☐
R	tinyint	☐
G	tinyint	☐
B	tinyint	☐
O	tinyint	☐
Data	nvarchar(MAX)	☐

شکل ۲ – جدول ایجاد شده *MyObject* در پایگاه داده

کنترل های مدیریت مدلControllers :

در این قسمت ما به تعریف کنترل ها و اکشن های مختلف هر مدل میپردازیم و با استفاده از آنها اطلاعات را مدیریت و هندل[11] میکنیم که هر کنترلر دارای چند دستور اصلی می باشد که شامل بخش ها زیر می باشد:

1-Index: در این اکشن لیست تمامی موجودیت های ذخیره در پایگاه اطلاعاتی را نمایش میدهد.

2-Create: در این اکشن ما میتوانیم برای هر موجودیت یک رکورد جدید در پایگاه داده ها تعریف و اضافه نماییم.

3-Details: در این اکشن ما میتوانیم جزئیات اطلاعات مرتبط با هر رکورد را مشاهده و بررسی نماییم.

4-Edit: در این اکشن ما میتوانم جزئیات اطلاعات مرتبط با هر رکورد را ویرایش کنیم و در پایگاه داده ها ذخیره نماییم.

5-Delete: در این اکشن ما میتوانیم رکورد اطلاعات مورد نظر را انتخاب و از پایگاه داده ها حذف نماییم.

کنترلر ObjectTypesController

در این قسمت ما به تعریف اکشن های مختلف این Controllerمی پردازم

۱-تعریف اکشن Index

در این قسمت ما با اتصال به دیتا بیس لیست تمامی Record های مرتبط به آبجکت تایپ ها مختلف را از Databaseدریافت به Viewمنتقل میکنیم.

قطعه کد زیر برای این قسمت مینویسیم:

```
// GET: ObjectTypes
3 references
public async Task<IActionResult> Index()
{
    return _context.ObjectType != null ?
                View(await _context.ObjectType.ToListAsync()) :
                Problem("Entity set 'VisualizationDataPanelContext.ObjectType'  is null.");
}
```

کد ۵ – اکشن جهت دریافت اطلاعات *ObjectType* ها از پایگاه داده

[11] Handle

۲-تعریف اکشن Details

در این قسمت ما با اتصال به دیتا بیس جزئیات Recordمورد نظر مرتبط به آبجکت تایپ را از Databaseدریافت به Viewمنتقل میکنیم.

```
// GET: ObjectTypes/Details/5
0 references
public async Task<IActionResult> Details(int? id)
{
    if (id == null || _context.ObjectType == null)
    {
        return NotFound();
    }

    var objectType = await _context.ObjectType
        .FirstOrDefaultAsync(m => m.ObjectTypeID == id);
    if (objectType == null)
    {
        return NotFound();
    }

    return View(objectType);
}
```

کد ۶ – اکشن جهت دریافت اطلاعات یک *ObjectType* از پایگاه داده

۳-تعریف اکشن Create

در این قسمت ما با اتصال به دیتا بیس یک Record جدید را به آبجکت تایپ ها در درون Databaseایجاد میکنیم.

```
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Create([Bind("ObjectTypeID,Name,Picture,URL")] ObjectType objectType)
{
    if (ModelState.IsValid)
    {
        _context.Add(objectType);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(objectType);
}
```

کد ۷– اکشن جهت ایجاد یک *ObjectType* جدید در پایگاه داده

۴-تعریف اکشن Edit

در این قسمت ما با اتصال به دیتا بیس یک Record مرتبط با آبجکت تایپ را در درون Databaseویرایش میکنیم.

```
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Edit(int id, [Bind("ObjectTypeID,Name,Picture,URL")] ObjectType objectType)
{
    if (id != objectType.ObjectTypeID)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        try
        {
            _context.Update(objectType);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!ObjectTypeExists(objectType.ObjectTypeID))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
        return RedirectToAction(nameof(Index));
    }
    return View(objectType);
}
```

کد ۸ – اکشن جهت ویرایش یک *ObjectType* در پایگاه داده

۵-تعریف اکشن Delete

در این قسمت ما با اتصال به دیتا بیس یک Record مرتبط با آبجکت تایپ را از در درون Databaseحذف میکنیم.

```
// POST: ObjectTypes/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> DeleteConfirmed(int id)
{
    if (_context.ObjectType == null)
    {
        return Problem("Entity set 'VisualizationDataPanelContext.ObjectType'  is null.");
    }
    var objectType = await _context.ObjectType.FindAsync(id);
    if (objectType != null)
    {
        _context.ObjectType.Remove(objectType);
    }

    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

1 reference
private bool ObjectTypeExists(int id)
{
  return (_context.ObjectType?.Any(e => e.ObjectTypeID == id)).GetValueOrDefault();
}
}
```

کد ۹ – اکشن جهت حذف یک *ObjectType* از پایگاه داده

کنترلر MyObjectController

در این قسمت ما به تعریف اکشن های مختلف این Controllerمی پردازم

۱-تعریف اکشن Index

در این قسمت ما با اتصال به دیتا بیس لیست تمامی Recordمرتبط به MyObjectتایپ ها مختلف را از Databaseدریافت به Viewمنتقل میکنیم.

قطعه کد زیر برای این قسمت مینویسیم:

```
// GET: MyObjects
3 references
public async Task<IActionResult> Index()
{
    var visualizationDataPanelContext = _context.MyObject.Include(m => m.objectType);
    return View(await visualizationDataPanelContext.ToListAsync());
}
```

کد ۱۰ – اکشن جهت دریافت اطلاعات *MyObject* ها از پایگاه داده

-2تعریف اکشن Details

در این قسمت ما با اتصال به دیتا بیس جزئیات Recordمورد نظر مرتبط به MyObjectرا از DataBaseدریافت به Viewمنتقل میکنیم.

```
// GET: MyObjects/Details/5
0 references
public async Task<IActionResult> Details(int? id)
{
    if (id == null || _context.MyObject == null)
    {
        return NotFound();
    }

    var myObject = await _context.MyObject
        .Include(m => m.objectType)
        .FirstOrDefaultAsync(m => m.MyObjectID == id);
    if (myObject == null)
    {
        return NotFound();
    }

    return View(myObject);
}
```

کد ۱۱ – اکشن جهت دریافت اطلاعات یک *MyObject* از پایگاه داده

-3تعریف اکشن Create

در این قسمت ما با اتصال به دیتا بیس یک Record جدید را به MyObjectدر درون DataBaseایجاد میکنیم.

```
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Create([Bind
("MyObjectID,Name,ObjectTypeID,PostionX,PostionY,PostionZ,RotationX,RotationY,RotationZ,ScaleX,ScaleY,ScaleZ,R,G,B,O,Data")]
MyObject myObject)
{
    if (ModelState.IsValid)
    {
        _context.Add(myObject);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    ViewData["ObjectTypeID"] = new SelectList(_context.ObjectType, "ObjectTypeID", "ObjectTypeID", myObject.ObjectTypeID);
    return View(myObject);
}
```

کد ۱۲– اکشن جهت ایجاد یک *MyObject* جدید در پایگاه داده

-4تعریف اکشن Edit

در این قسمت ما با اتصال به دیتا بیس یک Record موردMyObjectرا در درون DataBaseویرایش میکنیم.

```
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Edit(int id, [Bind
("MyObjectID,Name,ObjectTypeID,PostionX,PostionY,PostionZ,RotationX,RotationY,RotationZ,ScaleX,ScaleY,ScaleZ,R,G,B,O,Data")]
MyObject myObject)
{
    if (id != myObject.MyObjectID)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        try
        {
            _context.Update(myObject);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!MyObjectExists(myObject.MyObjectID))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
        return RedirectToAction(nameof(Index));
    }
    ViewData["ObjectTypeID"] = new SelectList(_context.ObjectType, "ObjectTypeID", "ObjectTypeID", myObject.ObjectTypeID);
    return View(myObject);
}
```

کد ۱۳ – اکشن جهت ویرایش یک *MyObject* در پایگاه داده

5–تعریف اکشن Delete

در این قسمت ما با اتصال به دیتا بیس یک Record موردMyObjectرا از در درون DataBaseحذف میکنیم.

```
// POST: MyObjects/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> DeleteConfirmed(int id)
{
    if (_context.MyObject == null)
    {
        return Problem("Entity set 'VisualizationDataPanelContext.MyObject'  is null.");
    }
    var myObject = await _context.MyObject.FindAsync(id);
    if (myObject != null)
    {
        _context.MyObject.Remove(myObject);
    }

    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

1 reference
private bool MyObjectExists(int id)
{
  return (_context.MyObject?.Any(e => e.MyObjectID == id)).GetValueOrDefault();
}
}
```

کد ۱۴ – اکشن جهت حذف یک *MyObject* از پایگاه داده

-6اکشن ایجاد فایل JSON MyObjectList

در این قسمت با تعریف این اکشن میتوانیم تمامی لیست آبجکت های تعریف شده در پایگاه داده MyObjectListرا به صورت یک فایل JOSNتبدیل و به صورت یک APIلیست برای بخش نمایش آماده نماییم.

کد این قسمت به صورت زیر نوشته میشود.

```
0 references
public async Task<IActionResult> MyObjectListAPI()
{
    var visualizationDataPanelContext = _context.MyObject;
    var getMyObject = await visualizationDataPanelContext.ToListAsync();

    var response = new MyObjectListResponse { MyObjectList = getMyObject };

    return Json(response);
    //return View();
}
```

کد ۱۵ – اکشن دریافت اطلاعات *MyObject* و تبدیل به فرمت *JSON*

بخش دوم شبیه سازی تجسم: Visualization Simulation

در این بخش، ما به طراحی و پیاده‌سازی شبیه‌سازی تجسم می‌پردازیم. به این معنا که با طراحی این قسمت، اطلاعات را از طریق وب APIدریافت کرده و در نرم‌افزار شبیه‌سازی نمایش می‌دهیم.

این بخش شامل دو قسمت مختلف زیر است:

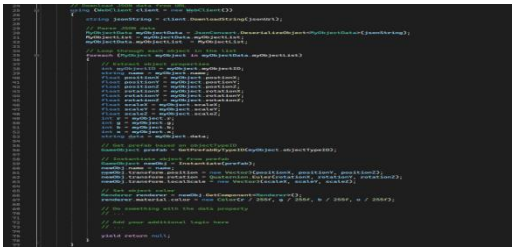
۱.نمایش آبجکت های ساخته شده

۲.نمایش تمام اطلاعات از آبجکت های ساخته شده

نمایش آبجکت های ساخته شده:

در این قسمت ما با میبایست قطعه کدی بنویسیم که با اتصال به وب پنل و با استفاده از تکنولوژی REST APIاطلاعات مرتبط با آبجکت های تعریف شده در پایگاه داده ها را دریافت و در محیط شبیه سازی شده به نمایش درآورد.

کد مورد نظر را به شیوه زیر در زبان C# می نویسیم:



کد ۱۶– دریافت اطلاعات از پنل و ایجاد ابجکت ها

تعریف آبجکت ها در پنل:

با توجه به اینکه تعداد مدل ها و اجسام سه بعدی بسیار زیاد می باشد و ما برای نمونه برای این مقاله این شکل مکعب یا همان Cube و کره یا Sphereرا آماده و داخل پنل وارد می نماییم.

کدگذاری داده‌ها:

ما در این بخش سیستم را به گونه ای طراحی و پیاده سازی کرده ایم که به صورت خودکار برای هر یک از اجسام سه بعدی که در پنل تعریف میکنیم به صورت خودکار یک کد منحصر به فرد ایجاد و هر جسم به صورت خودکار اختصاص دهد. و به آن در اصطلاح Primary Keyیا کلید اصلی می گوییم که معمولا با نام هر مدل و در انتها با کلمه IDبه پایان میرسد.

به عنوان مثال ما در این بخش پنل وب کلید های اصلی زیر را مد نظر قرار میدهیم:

۱. ObjectTypeID
۲. MyObjectID

نمایش داده‌ها :**Data Display**

در این قسمت، به بخش نمایش داده‌ها می‌پردازیم و قصد داریم دیتاهای ایجاد شده توسط وب پلتفرم را بررسی کنیم. برای این منظور، برای هر بخش،API های نوشته شده را فعال کرده و خروجی اطلاعات هر قسمت را بررسی می‌کنیم.

با فعال کردن API ها و استفاده از آنها، می‌توانیم به داده‌های ایجاد شده توسط وب پلتفرم دسترسی پیدا کنیم و این داده‌ها را بررسی و مورد ارزیابی قرار دهیم. با استفاده از خروجی‌های اطلاعات هر قسمت، قادر خواهیم بود تا تحلیل‌های مورد نیاز را بر روی داده‌ها انجام داده و به نتایج مورد انتظار دست پیدا کنیم.

نمایش داده :MyObjectAPI

در این بخش پس از فعال سازی این قسمت میتوانیم خروجی JSONرا مشاهده نماییم.


```
{
  "myObjectList": [
    {
      "myObjectID": 1,
      "name": "TestBox",
      "objectType": null,
      "objectTypeID": 1,
      "postionX": 0,
      "postionY": 0,
      "postionZ": 0,
      "rotationX": 0,
      "rotationY": 0,
      "rotationZ": 0,
      "scaleX": 1,
      "scaleY": 1,
      "scaleZ": 1,
      "r": 255,
      "g": 255,
      "b": 255,
      "o": 255,
      "data": "Box"
    },
    {
      "myObjectID": 2,
      "name": "Sphere",
      "objectType": null,
      "objectTypeID": 2,
      "postionX": 2,
      "postionY": 0,
      "postionZ": 0,
      "rotationX": 0,
      "rotationY": 0,
      "rotationZ": 0,
      "scaleX": 1,
      "scaleY": 1,
      "scaleZ": 1,
      "r": 255,
      "g": 0,
      "b": 0,
      "o": 100,
      "data": "0"
    }
  ]
}
```


شکل ۳ – خروجی *API MyObject* به صورت *JSON*

با مشاهده فایل JSONخروجی MyObjectAPI، متوجه می‌شویم که اطلاعاتی که توسط این اکشن ایجاد شده‌اند، دقیقاً با اطلاعاتی که در پایگاه داده‌ها ذخیره شده‌اند، برابر هستند و به درستی به فرمت استاندارد JSONتبدیل شده‌اند.

تجزیه و تحلیل داده‌ها:

در این بخش به تحلیل و تجزیه‌ی داده‌ها می‌پردازیم و آن را به دو دسته‌بندی اصلی تقسیم می‌کنیم: تحلیل کمی و تحلیل کیفی. این دو دسته‌بندی به تفصیل در زیر توضیح داده شده است.

تجزیه و تحلیل کمی:

در این بخش، ما به تحلیل کمی اطلاعات دریافتی می‌پردازیم و بررسی می‌کنیم آیا اطلاعات ذخیره شده در پایگاه داده‌ها و فایل JSONسمت وب پنل با اطلاعات دریافتی در بخش Visualization Simulation همخوانی دارند. هدف ما در اینجا این است که اطمینان حاصل کنیم که هیچ تداخلی در این فرآیند وجود ندارد و اطلاعات به صورت کامل و درست بین دو بخش مختلف منتقل شده‌اند.

تجزیه و تحلیل کمی قسمت MyObjects

در این بخش، پس از فعالسازی قسمت Visualization Simulation، سیستم به وب پنل متصل می‌شود و درخواست دریافت اطلاعات را ارسال می‌کند. سپس نتیجه اطلاعات دریافتی را در کنسول چاپ می‌کند.

به طور واضح مشاهده می‌شود که تمام اطلاعات به درستی به این قسمت منتقل شده‌اند و لیست تمام آبجکت‌های تعریف شده در وب پنل نیز به درستی در کنسول MyObjects چاپ شده است.

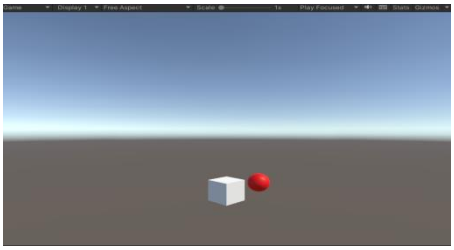
تجزیه و تحلیل کمی: MyObjects

در این بخش، ما قصد داریم اطلاعات ایجاد و منتقل شده را از نظر کیفی مورد بررسی قرار دهیم. هدف ما در اینجا، مشاهده ویژوالی تمام اطلاعاتی است که در پایگاه داده‌ها و وب پنل تعریف شده‌اند. برای این منظور، برای

هر یک از APIهای نوشته شده به صورت جداگانه، این مورد را بررسی می‌کنیم.

تجزیه و تحلیل کیفی: MyObjects

در این قسمت، پس از اجرای بخش "Visualization Simulation"، باید تمامی آبجکت‌های تعریف شده در بخش "MyObject" را در محیط سه بعدی مشاهده کنیم. برای این منظور، بخش شبیه‌سازی را اجرا می‌کنیم و به نتیجه زیر می‌رسیم.



شکل ۴ – نمایش نمونه های ایجاد شده نرم افزار شبیه ساز

همانطور که مشاهده می‌شود، دو آبجکت تعریف شده در پنل به خوبی و به درستی در این قسمت نمایش داده می‌شوند.

نتایج:

در این بخش، ما به جمع‌آوری اطلاعاتی که به صورت اعداد هستند، می‌پردازیم. به این منظور، اطلاعات جمعیت مناطق یک شهر مانند تهران را جمع‌آوری و مورد بررسی قرار می‌دهیم.

شهر تهران با داشتن ۲۲ منطقه، اطلاعات جمعیتی ساکنان در شهر تهران را براساس آخرین آمار و اطلاعات جمع‌آوری شده توسط مرکز آمار ایران در سال ۱۳۹۵ در جدول زیر مشاهده می‌کنید. ما این اطلاعات را استخراج می‌کنیم.

شکل ۵ – نمایش اطلاعات جمعیت مناطق ۲۲ گانه تهران

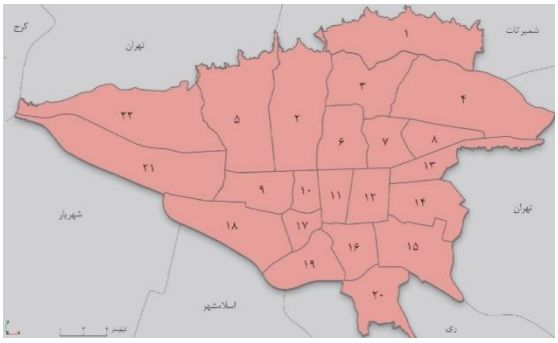
بر اساس شکل شماره ۵ اطلاعات را زیر را استخراج و به شکل جدول زیر در میاوریم. سپس اطلاعات جمعیت را به عنوان پارامتر Y در بخش Scale ها در نظر خواهیم گرفت.

شماره منطقه	تعداد جمعیت ساکن
1	**508،487**
2	**303،701**
3	**649،330**
4	**001،919**
5	**346،858**
6	**384،251**
7	**149،312**
8	**197،425**
9	**239،174**
10	**115،327**
11	**940،307**
12	**831،241**
13	**952،248**
14	**795،515**
15	**279،641**
16	**406،286**

17	**231،273**
18	**882،419**
19	**027،261**
20	**259،365**
21	**821،186**
22	**347،176**

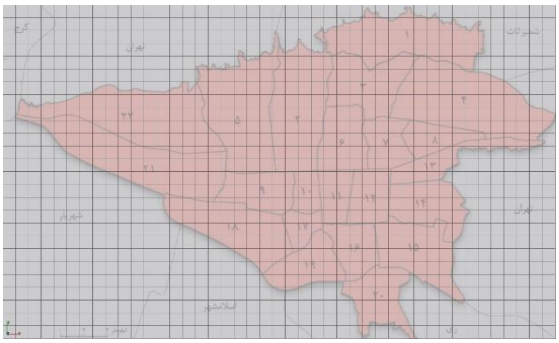
جدول ۱ – اطلاعات جمعیتی هر منطقه بر اساس تعداد نفرات ساکن

برای اینکه بتوانیم اطلاعات جمعیتی و در صورت نیاز سایر اطلاعات مرتبط با مناطق مختلف شهر را به صورت سه بعدی نمایش دهیم، نیازمند ایجاد یک ساختار نقشه تهران با تقسیم‌بندی به مکعب‌های مختلف براساس موقعیت جغرافیایی آن‌ها و دسته‌بندی آن‌ها بر اساس هر منطقه هستیم. برای این کار، از نقشه زیر استفاده می‌کنیم.



شکل ۶- نقشه مناطق۲۲ گانه تهران

در مرحله بعد، این نقشه را با استفاده از مکعب‌های کوچک پرچیدمان می‌کنیم تا تمام سطح نقشه با مکعب‌های مختلف پوشیده شود. این فرایند در شکل زیر مشاهده می‌شود.



شکل ۷ – تقسیم بندی نقشه تهران به مکعب های استاندارد

در مرحله بعد، بر اساس هر منطقه، مکعب‌ها را جداسازی می‌کنیم و به هر منطقه یک رنگ مشخص اختصاص می‌دهیم تا آنها به راحتی قابل تشخیص باشند. این عملیات در نمونه زیر قابل مشاهده است.

شکل ۸ – تغییر رنگ هر منطقه از شهر تهران با یک رنگ مختلف

سپس با حذف مکعب‌های اطراف، به شکلی می‌رسیم که نقشه شهر به صورت واضح‌تر قابل مشاهده است. در نتیجه، تصویر زیر نقشه شهر را نشان می‌دهد.

شـــکل ۹ – نمایش مناطق ۲۲ گانه شـــهر با مکعب ها در رنگ های مختلف

پس از به دســـت آوردن مختصـــات تمامی مکعب‌های ایجاد شـــده برای مناطق مختلف شـــهر، ما آغاز به وارد کردن تمامی این مختصات به پنل می‌کنیم.

در این نمونه، تعداد ۵۷۶ مکعب مورد نیاز اســـت که باید اطلاعات مکعب‌ها را به درســـتی به پنل وارد کنیم تا بتوانیم این اطلاعات را در پایگاه داده ذخیره کنیم.

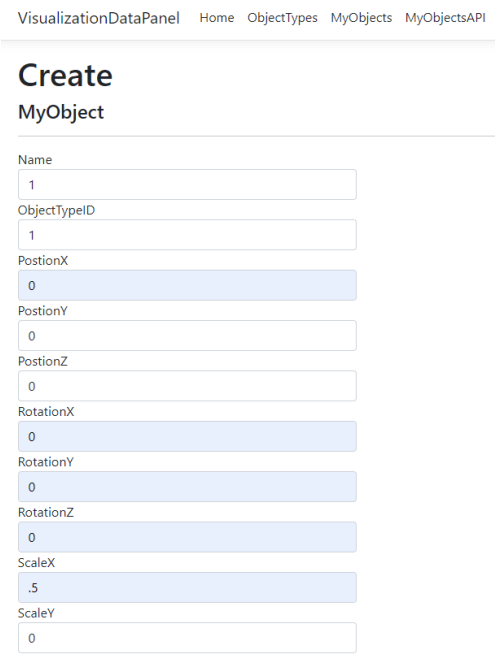
برای ورود اطلاعـات بـه منوی MyObjectوارد می‌شویم و گزینه Createرا انتخاب می‌کنیم.

سپس اطلاعات مرتبط با هر یک از این مکعب‌ها را در فیلدهای مربوطه وارد می‌کنیم.

این اطلاعات شـامل نام، نوع ابجکت، مختصـات X، Z،Y ، چرخش Z، Y،X ، رنگ هر مکعب به صـــورت RGBAو داده‌های مرتبط با هر منطقه شهری می‌باشند.

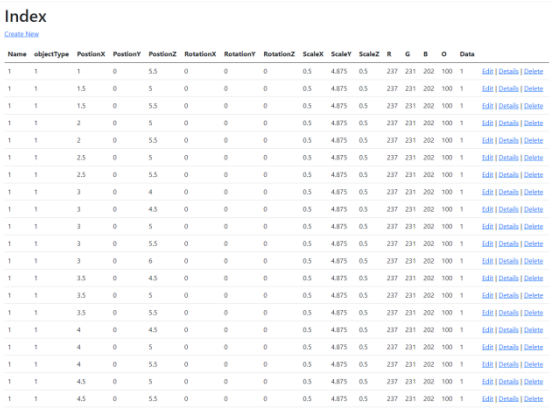
این اطلاعات به ترتیب وارد پنل می‌شوند.

شـــکل شـــماره ۱۰ نحوه ورود این اطلاعات به پنل را نشان می‌دهد



شکل ۱۰ – پنل ورود اطلاعت آبجکت ها

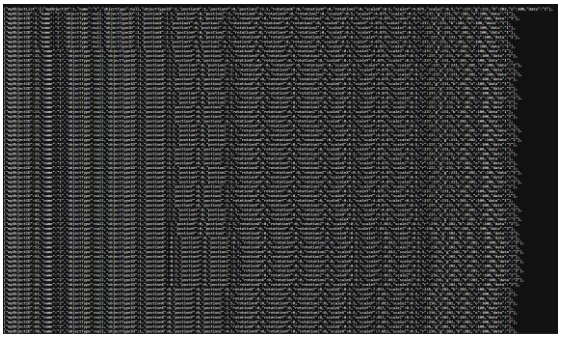
برای هر مکعب، مطابق شـــکل بالا، اطلاعات را وارد می‌کنیم. سپس پس از ورود تمامی اطلاعات، می‌توانیم در بخش indexنتیجه را مشـاهده کنیم که تمامی اطلاعات مکعب‌ها به درستی در پنل ثبت شده‌اند، مطابق شکل زیر



شـــکل ۱۱ – پنل پس از ورود تمامی اطلاعت مرتبط با مکعب ها

حال که اطمینان حاصـل کردیم که اطلاعات به طور کامل در پـایگـاه داده ثبـت شـــــده‌انـد، می‌توانیم بـه بخش MyObjectListAPIرفته و اطمینان حاصـــل کنیم که

تمام اطلاعات ذخیره شده در پایگاه داده به فرمت استاندارد JSONتبدیل شده‌اند، مانند شکل زیر:

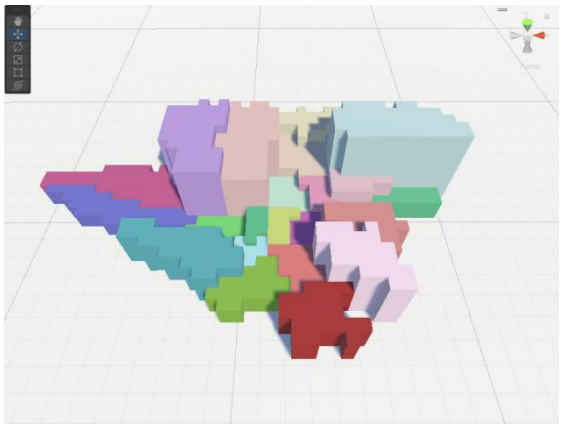


شکل ۱۲ – خروجی *JSON MyObjectAPI*

پس از این مرحله، وارد نرم‌افزار Visualization Simulationمی‌شویم تا ارتباط صحیح بین نرم‌افزار شبیه‌سازی و پنل را تضمین کنیم. برای این منظور، پس از اجرای نرم‌افزار، همانطور که مشاهده می‌شود، ارتباط به درستی برقرار می‌شود و تمامی اطلاعات مرتبط با آبجکت‌ها از سمت سرور به کلاینت و نرم‌افزار شبیه‌سازی به صورت کامل و درست منتقل می‌شوند. سپس کلاینت تمامی اطلاعات را به طور کامل دریافت کرده و نتیجه را در کنسول به شکل زیر چاپ می‌کند:

شکل ۱۳– دریافت اطلاعات ***JSON*** از پنل و چاپ در کنسول

در این مرحله، برای مشاهده و نمایش خروجی به صورت سه بعدی، به قسمت نمایش و مشاهده می‌پردازیم تا اطمینان حاصل کنیم که معکب‌ها و پارسل‌ها در نرم‌افزار شبیه‌ساز Visualization Simulation به درستی ایجاد شده‌اند. با مراجعه به این قسمت، به نتیجه زیر می‌رسیم و می‌بینیم که این نرم‌افزار تمامی اطلاعات دریافتی را تک به تک بررسی کرده و بر اساس اطلاعات دریافتی از سمت سرور (که به صورت فایل JSON دریافت شده است)، تمامی پارسل‌ها و معکب‌های اطلاعاتی را در فضای سه بعدی ترسیم می‌کند. نتیجه آن به شکل زیر قابل مشاهده است:



شکل ۱۴– شبیه سازی خروجی نهایی از اطلاعات ثبت شده در پنل

۴- بحث و نتیجه گیری:

در این مقاله، پس از مشاهده نتایج، به این نتیجه می‌رسیم که این نرم‌افزار و پلتفرم قادر است اطلاعات چند بعدی را به شیوه‌های متفاوت و جدید به نمایش بگذارد. همچنین، می‌تواند چندین عنصر و شاخص مختلف را در یک تصویر شبیه‌سازی شده نمایش دهد. در این مثال، علاوه بر نمایش مناطق مختلف شهری به رنگ‌های مختلف، می‌توانیم اطلاعات جمعیتی مناطق مختلف را نیز به شکل کاملاً متفاوت و خلاقانه به نمایش بگذاریم، به عنوان مثال با استفاده از ارتفاع سه‌بعدی. همچنین، این امکان وجود دارد که این اطلاعات به صورت دینامیک به‌روزرسانی شده و اطلاعات دیگری را به این خروجی متصل کنیم.

میتوانیم با استفاده از این پلتفرم و توسعه آن به قابلیت های زیر برسیم.

۱. ایجاد یک مدل استاندارد برای نمایش اطلاعات به شیوه دلخواه پژوهشگران.
۲. امکان ورود آسان و ساده اطلاعات به نرم‌افزار بدون نیاز به دانش فنی و پیچیدگی‌های کامپیوتری و بدون نیاز به کد نویسی و نرم‌افزارهای خاص.
۳. قابلیت آسان و تغییرپذیری اطلاعات درون پنل و به‌روزرسانی خودکار آن‌ها.
۴. قابلیت افزودن اطلاعات مختلف به‌صورت لحظه‌ای.
۵. قابلیت اتصال اطلاعات نمایش به منابع مختلف و ایجاد شبیه‌سازی دینامیک و پویا به‌صورت خودکار.
۶. قابلیت ایجاد ارتباطات بین اجزاء مختلف در شبیه‌سازی.
۷. امکان تصویرسازی و نمایش گرافیکی پیشرفته.
۸. قابلیت تحلیل و استخراج داده‌های پیچیده از اطلاعات.
۹. قابلیت انجام آزمایش‌های تعاملی و تغییرپذیری پارامترها در زمان واقعی.
۱۰. قابلیت به‌روزرسانی مداوم نرم‌افزار با افزودن قابلیت‌های جدید و بهبودهای اصلاحی.

در نتیجه، این پلتفرم و نرم‌افزار به پژوهشگران این امکان را می‌دهد که به‌صورت آسان و سریع، اطلاعات با حجم زیاد و پیچیدگی‌های مختلف را به نمایش بگذارند و به‌راحتی آن‌ها را تحلیل کنند. پیشنهاد می‌شود در آینده، به این نرم‌افزار قابلیت‌های بیشتری اضافه شود، به عنوان مثال، امکان دریافت اطلاعات به‌صورت Real-Time و نمایش آخرین اطلاعات دریافتی از سرورهای مختلف.

•

۵- منابع:

۱-گرجی، ابراهیم و برخورداری، سجاد؛ "مبانی روش تحقیق در علوم اجتماعی"، تهران، ثالث، (۱۳۸۸) صفحه ۳۳

۲- ایران نژاد، پاریزی ،مهدی، "روش های تحقیق در علوم اجتماعی"، تهران: نشر مدران، (۱۳۸۲) صفحه ۵۰

۳- الوانی، مهدی، "تصمیم گیری و تعیین خط مشی دولتی "،سازمان مطالعه و تدوین کتب علوم انسانی سمت (۱۳۷۹) صفحه ۵۴

۴-رزاقی، افشین، " نظریه‌های ارتباطات اجتماعی"، تهران، پیکان (۱۳۸۱) صفحه ۱۸۴

۵-فرهنگی، علی اکبر. صفرزاده، حسین "روشهای تحقیق در علوم انسانی با نگرشی بر پایان نامه نویسی"، تهران انتشارات (۱۳۸۵) صفحه ۶۹

۶-حافظ نیا، محمد،" مقدمه ای بر روش تحقیق در علوم انسانی" انتشارات سمت (۱۳۸۷) صفحه ۱۶

۹- گروه مؤلفین تحت نظر لرنر، ادونس؛ فرهنگ آکسفورد، تهران، نشر ثامن‌الائمه (علیه‌السلام) و نشر منادی، ۱۳۸۴، چاپ اول، ص ۱۲۰۰

۱۲-.شانون، رابرت؛ علم و هنر شبیه‌سازی سیستم‌ها، علی‌اکبر عرب مازار، تهران، مرکز نشر دانشگاهی، ۱۳۸۰، چاپ دوم، ص۲۰

۱۳-اخوان، پیمان و معینی، علیرضا؛ نقش مهندسی مجدد در شبیه‌سازی سازمان، مجله تدبیر، ۱۳۸۴، شماره ۱۵۸، سال سیزدهم، ص۱۶.

14-رنجگری، علیرضا؛ شبیه‌سازی ابزاری نیرومند برای تصمیم‌گیری، مجله روشن، ۱۳۷۷، شماره۴۷، ص۲۹

16-صالح فتح‌آبادی، حسن؛ شبیه‌سازی سیستم‌ها به‌وسیله کامپیوترهای رقمی، تهران، نشر جهاد دانشگاهی، ۱۳۶۵، چاپ اول، ص۱۰۱

منابع لاتین:

7-Editorial board: Graphical Models, Elsevier, retrieved 2021-10-03

8-Nixon, Mark (2013), "1.6.1 Journals and magazines", Feature Extraction and Image Processing, Elsevier, pp. 24–25, ISBN 9780080506258

10-Merriam; Webster's Collegiate Dictionary, Tehran, Ebteda, 2000, 10th ed, P1094.

11-FERRIN D.M, MOTHLER, Did. & MUTHLER, M.J; Six sigma and simulation, so what's the Correlation? SIMULATION Conference Manchester, 2002, P1439-43.

15-Keller, M.I. & Raymond, J.M. & Raffo, D.M; Software process simulation modeling: Way? What? How? Journal of system and software, 1999, Vol.46 Nos.2/3, P254-59

17- "evl – electronic visualization laboratory". www.evl.uic.edu. Archived from the original on 30 April 2020. Retrieved 2 September 2018.

18- Michael Friendly (2008). "Milestones in the history of thematic cartography, statistical graphics, and data visualization". Project moved to http://datavis.ca/milestones/

19- "Scientific Visualization." sciencedaily.com. Science Daily, 2010. Retrieved from web https://www.sciencedaily.com/articles/s/scientific_visualization.htm Wayback Machine. on 17 November 2011.

20- "Scientific Visualization." Scientific Computing and Imaging Institute. Scientific Computing and Imaging Institute, University of Utah, n.d. Retrieved from web http://www.sci.utah.edu/research/visualization.html. on 17 November 2011.

21- Opiła, Janusz (1 April 2019). "Role of Visualization in a Knowledge Transfer Process". Business Systems Research Journal. 10 (1): 164–179. doi:10.2478/bsrj-2019-0012. ISSN 1847-9375.

22- Opila, J.; Opila, G. (May 2018). "Visualization of computable scalar 3D field using cubic interpolation or kernel density estimation function". 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). Opatija: IEEE: 0189–0194. doi:10.23919/MIPRO.2018.8400036. ISBN 9789532330953.

23- Thomas, J.J., and Cook, K.A. (Eds) (2005). An Illuminated Path: The Research and Development Agenda for Visual Analytics, IEEE Computer Society Press