



دانشگاه صنعتی نوشیروانی بابل

دانشکده مهندسی برق و کامپیوتر

پروژه کارشناسی

در رشته مهندسی کامپیوتر گرایش نرم افزار

عنوان:

اپلیکیشن رای گیری های دانشگاه (انجمن ها و شورای سنفی دانشجویان)

دانشجویان:

محمدعلی کیه

استاد راهنما:

دکتر مهدی عمادی

بهمن ۱۴۰۲

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

چکیده

این پروژه سیستم رای گیری انجمن‌های دانشجویی را معرفی می‌کند. این نرم افزار امکان رای گیری‌های مختلفی که در سطح دانشگاه انجام می‌شود را برای دانشجویان فراهم می‌سازد و با بکار گیری مکانیزم‌های مختلف امنیتی مناسب و بهینه از هرگونه مشکل امنیتی و حفاظتی جلوگیری می‌کند. همچنین، این نرم افزار امکان بهبود کاربرد را نیز داراست و در صورت نیاز، توانایی تبدیل شدن به یک نرم افزار جامع برای مدیریت امور مختلفی که یک دانشجو در دانشگاه نیاز دارد را می‌تواند فراهم کند چرا که سعی بر پیاده سازی نرم افزار در پلتفرم‌های مختلف انجام شده است و دانشجو به راحتی می‌تواند با استفاده از تلفن همراه یا رایانه شخصی خود به آن دسترسی داشته و از مراحل پیچیده و طولانی ورود به سامانه‌هایی که عمل مشابه را انجام می‌دهند، فارغ باشد.

رای گیری‌های سنتی ممکن است با مشکلاتی همچون رای گیری مکرر، تقلب و اشتباهات انسانی همراه باشد. ایجاد یک نرم افزار رای گیری اختصاصی، تسهیل مشارکت دانشجویان در فرآیند تصمیم گیری و شفافیت بیشتر در رای گیری‌ها را به ارمغان می‌آورد. امیدواریم این پروژه بهبودی اثربخش در فرآیند تصمیم گیری‌ها و عملکرد انجمن‌های دانشجویی به ارمغان بیاورد و مورد استفاده و استقبال گسترده‌ای از اعضا و دانشجویان قرار گیرد.

واژگان کلیدی: رای گیری، انجمن دانشجویی، نرم افزار، امنیت

فهرست

چکیده	۳
فصل اول : کلیات	۸
۱-۱ مقدمه، ضرورت و کاربرد	۹
۱-۲ تعریف مسئله	۱۰
فصل دوم : ادبیات گزارش	۱۱
۲-۱ مقدمه	۱۲
۲-۲ بک اند	۱۲
۲-۳ پایگاه داده	۱۲
۲-۴ API	۱۳
۲-۵ اندپوینت	۱۳
۲-۶ کتابخانه	۱۳
۲-۷ فریمورک	۱۳
فصل سوم: ابزار، تکنولوژی ها و کتابخانه ها	۱۴
۳-۱ مقدمه	۱۵
۳-۲ Django	۱۶
۳-۳ Django Rest Framework (DRF)	۱۷
۳-۴ djoser	۱۸
۳-۵ JWT	۱۸
۳-۶ Git	۱۸
۳-۷ VS Code	۱۹

۱۹Visual Paradigm ۳-۸
۲۰ فصل چهارم: طراحی و ساختار
۲۱ ۴-۱ طراحی پایگاه داده
۲۱ ۴-۱-۱ مدل ERD سیستم
۲۱ ۴-۱-۲ User. جدول
۲۳ ۴-۱-۳ Representative. جدول
۲۴ ۴-۱-۴ votecart. جدول
۲۵ ۴-۱-۵ votes. جدول
۲۶ ۴-۲ احراز هویت
۲۶ ۴-۳ . تصاویر نهایی
۲۸ فصل پنجم: پیاده سازی
۲۹ ۵-۱ مقدمه
۲۹ ۵-۲ کنترلر ادمین
۲۹ ۵-۲-۱ ورود
۲۹ ۵-۲-۲ ثبت نام و لیست ادمین ها
۳۰ ۵-۲-۳ پروفایل و ویرایش اطلاعات ادمین
۳۰ ۵-۳ کنترلر دانشجو
۳۰ ۵-۳-۱ ثبت نام
۳۰ ۵-۳-۲ ورود
۳۱ ۵-۳-۳ پروفایل

۳۱	۵-۴ کنترلر کاندید
۳۱	۵-۴-۱ ثبت نام، ویرایش و لیست همه نامزدها.....
۳۲	۵-۵ کنترلر انتخابات
۳۲	۵-۵-۱ ثبت انتخابات ، لیست همه انتخابات ها.....
۳۲	۵-۵-۲ نتیجه یک انتخابات برای ادمین
۳۳	۵-۵-۳ لیست همه انتخابات های دانشجو
۳۳	۵-۵-۴ لیست نامزدهای یک انتخابات.....
۳۴	۵-۶ کنترلر رای
۳۴	۵-۶-۱ ثبت رای
۳۵	۵-۶-۷ خروجی نهایی دیتابیس در نرم افزار power bi
۳۶	فصل ششم: نتیجه گیری و پیشنهادها
۳۸	مراجع

فصل اول : کلیات

۱-۱ مقدمه، ضرورت و کاربرد

رای گیری انجمن های دانشجویی هرساله توسط مسئولان دانشگاه بصورت حضوری و غیر الکترونیکی انجام می شود.

این نوع از رای گیری ها در انجمن ها و اجتماعات دانشجویی به منظور تسهیل فرآیند تصمیم گیری ها، تعیین سیاست ها، انتخابات مختلف و همچنین اعمال تغییرات مورد استفاده قرار می گیرند. با توجه به اهمیت این ابزار، ایجاد نرم افزاری که بتواند این فرآیند را به صورت دقیق، شفاف و موثر پشتیبانی کند، بسیار مورد نیاز است.

هدف این پروژه طراحی و پیاده سازی یک نرم افزار رای گیری اختصاصی برای انجمن های دانشجویی است که امکان رأی گیری در مسائل مختلف را فراهم می سازد. این نرم افزار از ویژگی ها و قابلیت های گوناگونی برخوردار خواهد بود که به دانشجویان و اعضای انجمن ها امکان مشارکت فعال در فرآیند تصمیم گیری ها را می دهد. همچنین، این نرم افزار برای افزایش شفافیت و اعتماد در فرآیند رأی گیری، از مکانیزم های امنیتی و حفاظتی نیز بهره میبرد.

دلایل انجام این پروژه به دو دسته مهم تقسیم می شود:

نیاز به اتوماسیون فرآیند رای گیری: در رای گیری انجمن های دانشجویی، فرآیند رأی گیری به صورت سنتی انجام می شود که ممکن است با مشکلاتی همچون رای گیری مکرر، تقلب و اشتباهات انسانی همراه باشد. با ایجاد یک نرم افزار رای گیری متمرکز و دقیق، امکان اتوماسیون فرآیند رأی گیری فراهم می شود که باعث کاهش این مشکلات خواهد شد.

تسهیل مشارکت و شفافیت: انجمن های دانشجویی به عنوان نهادهای مهم در محیط دانشگاهی، باید دانشجویان را به مشارکت فعال و مستمر ترغیب کنند. ایجاد یک نرم افزار رای گیری با امکانات کاربردی و رابط کاربری موثر، می تواند تعامل دانشجویان را با فرآیند رأی گیری بهبود بخشد و شفافیت بیشتری را در این فرآیند فراهم آورد.

۱-۲ تعریف مسئله

انجمن های دانشجویی به عنوان نهاد های حیاتی در دانشگاه ها و محیط های آموزشی، نقش مهمی در تصمیم گیری ها، انتخابات و تغییرات مختلف دارند. رای گیری در این انجمن ها به منظور تعیین اعضا و رتبه بندی آنها مورد استفاده قرار می گیرد و برای شرکت کنندگان اهمیت بالایی دارد. اما رای گیری های سنتی که به صورت دستی و با استفاده از کاغذ انجام میشوند، ممکن است با مشکلاتی همچون عدم دقت، تقلب، از دست دادن برگه ها و همچنین پرهزینه بودن همراه باشند.

همچنین، انجمن های دانشجویی نیاز دارند تا فرآیند تصمیم گیری ها و رأی گیری ها به صورت شفاف و محکم انجام شود تا اعضا به نتایج اعتماد داشته باشند و بهره‌وری افزایش یابد. لازمه ی موفقیت انجمن های دانشجویی، ایجاد فرآیند رای گیری مناسب و کارآمد است که توانایی پشتیبانی از تصمیم گیری ها، اجرای انتخابات، انتشار نتایج و مدیریت رای گیری ها را داشته باشد.

بنابراین، با توجه به مسائل و مشکلات موجود در رای گیری های سنتی و نیاز به تسهیل مشارکت دانشجویان و افزایش شفافیت، ایجاد یک نرم افزار رای گیری اختصاصی برای انجمن های دانشجویی امری ضروری و کارآمد به نظر میرسد. این نرم افزار با استفاده از فناوری های مدرن و مکانیزم های امنیتی، امکان رأی گیری امن و دقیق را برای افراد فراهم می کند و تسهیلاتی را در مدیریت و اجرای رای گیری ها ایجاد می نماید.

فصل دوم : ادبیات گزارش

۲-۱ مقدمه

برای توسعه نرم افزار مطرح شده در این گزارش از مجموعه ای از تکنولوژی های تحت وب استفاده شده است که در این بخش به بررسی برخی اصطلاحات استفاده شده در گزارش پرداخته می شود.

۲-۲ بک اند

بک اند (Back-End) یکی از مفاهیم اصلی توسعه وب محسوب می شود و به بیان ساده بیان گر تمام اتفاقی است که در پشت صحنه یک وب سایت یا یک وب اپلیکیشن رخ می دهد. بر خلاف فرانت اند که به توسعه «جلو» یا ظاهر نمایشی برنامه اشاره دارد، بک اند به بخش «پشتی» یا همان بخش عملکردی برنامه می پردازد. این بخش به سمت سرور نزدیک تر است و کاربر به طور مستقیم به آن دسترسی ندارد، جایی که منطق، کدهای برنامه، پایگاه داده، وظایف مدیریتی و به طور کلی عملیات این چنینی انجام می شود. به طور مختصر و مفید، بک اند یک برنامه را می توان قلب تپنده یا نیروی محرکه آن دانست و تمام مواردی را در بر می گیرد که از دید کاربر پنهان است. این بخش، پاسخ به درخواست های کاربر و ارسال اطلاعاتی از بک اند به فرانت انت برنامه به منظور نمایش آن را انجام می دهد.

۲-۳ پایگاه داده

بانک اطلاعاتی یا همان پایگاه داده (دیتابیس | Database) مجموعه ای سازمان یافته از داده ها است. پایگاه های داده از ذخیره سازی الکترونیکی و ایجاد تغییر در داده ها پشتیبانی می کنند. مدیریت داده ها به وسیله پایگاه داده بسیار آسان می شود. برای مدیریت داده ها در یک بانک اطلاعاتی از سیستم مدیریت پایگاه داده (Database Management Syste) یا همان DBMS استفاده می شود. نمونه های معروف از پایگاه های داده شامل PostgreSQL، Microsoft SQL Server، Oracle، MySQL و MongoDB می باشند. هر نوع پایگاه داده ممکن است مدل های مختلفی از مدیریت داده ها را پشتیبانی کند، از جمله مدل رابط های (Relational)، مدل شیء گرا (Object-Oriented) و مدل های غیررابطه ای (Non-Relational). (Non) مانند پایگاه ای داده NoSQL.

۴-۲ API

API (Application Programming Interface) در واقع یک مجموعه از دستورالعمل ها و قوانینی است که توسط نرم افزارها به کار می رود تا با یکدیگر ارتباط برقرار کنند و امکان اشتراک داده ها و عملکردها را فراهم آورند. این رابط کاربری به برنامه نویس ها امکان می دهد تا از قابلیت ها و سرویس های یک سیستم، برنامه یا سرویس خارجی بدون اینکه به جزئیات پیچیده ترین عملکردهای آن وارد شوند بهره برداری کنند.

۵-۲ اندپوینت

اندپوینت (End point) به معنای نقطه پایانی است و به عنوان یک مفهوم کلیدی در طراحی API ها و وب سرویس ها مورد استفاده قرار می گیرد. اندپوینت نشان دهنده مکان و مسیری است که از طریق آن می توانید با یک سرویس یا منبع مشخص ارتباط برقرار کنید و اطلاعات مربوط به آن را دریافت یا ارسال کنید.

در معماری و طراحی API ها، هر عملیات مشخص که می خواهید انجام دهید (مثلاً دریافت لیست کاربران، ایجاد کاربر جدید، دریافت اطلاعات یک محصول و غیره)، یک اندپوینت متناظر دارد که این اعمال را برای شما انجام می دهد.

۶-۲ کتابخانه

کتابخانه (Library) در برنامه نویسی مجموعه کدهایی از قبل نوشته شده گفته می شود. کتابخانه به بسته بندی توابعی کمک می کند که در یک فایل باهم در ارتباط هستند و سازماندهی و دسترسی به این توابع از طریق کتابخانه آسان تر می شود. در دنیای برنامه نویسی و رو به پیشرفت امروزی، هر روز تعداد بسیاری برنامه نوشته می شوند و هر لحظه تعداد کدهای برنامه به صورت نمایی در حال افزایش است. کتابخانه در برنامه نویسی به جهت مدیریت چندین هزار و حتی میلیون ها خط از کدها مورد استفاده قرار می گیرد.

به علاوه، کتابخانه در برنامه نویسی برای استفاده مجدد از رفتارها توسط چندین برنامه مستقل ارائه شده است. یک برنامه، رفتار ارائه شده توسط کتابخانه را به وسیله زبان برنامه نویسی نشان می دهد.

۷-۲ فریمورک

فریمورک (Framework) یک چارچوب به حساب می آید که معمولاً برنامه نویس ها برای توسعه و طراحی نرم افزار از آن استفاده می کنند. استفاده از فریمورک به ساده شدن توسعه پردازش ها کمک می کند. یکی از دلیل های این سادگی این موضوع است که فریمورک از نوشتن مکرر کدهای مشابه جلوگیری خواهد کرد. همچنین باعث ایجاد شیوه های مناسب و سازگاری در کدها می شود. بنابراین می توان گفت که فریمورک، مجموعه ای از ابزارها و ماژول های گوناگون است که می توانند در پروژه های مختلف مورد استفاده قرار بگیرند.

فصل سوم: ابزار، تکنولوژی ها و کتابخانه ها

۳-۱ مقدمه

استفاده از تکنولوژی ها و کتابخانه ها در طراحی نرم افزار کارایی و کیفیت نرم افزارها را تضمین می کند و فرآیند توسعه را سرعت می بخشد. مهم ترین دلایل استفاده از تکنولوژی ها و کتابخانه ها عبارتند از:

انجام عملیات پیچیده: نرم افزارها ممکن است نیاز به انجام عملیات پیچیده و پیشرفته داشته باشند که پیاده سازی آنها از ابتدا به صورت کامل زمان بر و هزینه بر است. با استفاده از تکنولوژی ها و کتابخانه ها، می توان این عملیات ها را با سهولت و سرعت بیشتری انجام داد.

ایجاد کد قابل نگهداری: استفاده از تکنولوژی ها و کتابخانه ها به کمک طرح ها و معماری های موجود، کدی قابل نگهداری و قابل اطمینان ایجاد میکند. این امر به طور مستقیم باعث کاهش هزینه ها و زمان مورد نیاز برای توسعه و تست نرم افزار می شود.

افزایش کیفیت و عملکرد: تکنولوژی ها و کتابخانه ها ابزارهای موثری برای انجام تست های مختلف و بهبود کیفیت نرم افزارها هستند. از ابزارها و کتابخانه های مرتبط با تست و اعتبارسنجی می توان برای پیدا کردن خطاها و مشکلات در نرم افزارها استفاده کرد و بدین ترتیب کیفیت نهایی نرم افزار را افزایش داد.

انعطاف پذیری و توسعه پذیری: با استفاده از تکنولوژی ها و کتابخانه ها، نرم افزارها انعطاف پذیرتر و قابلیت توسعه بیشتری خواهند داشت. این امکان را به برنامه نویس ها می دهد تا با سرعت و کمترین تغییرات، امکانات و ویژگی های جدید را به نرم افزار اضافه کنند.

کاهش هزینه ها: استفاده از تکنولوژی ها و کتابخانه ها ممکن است به صورت رایگان یا با هزینه کمتری ارائه شود که باعث کاهش هزینه های توسعه و پشتیبانی نرم افزارها می شود.

به طور کلی، استفاده از تکنولوژی ها و کتابخانه ها در توسعه نرم افزارها، مزایای بسیاری را به همراه دارد و این امر به طور گستردهای در جلوگیری از تکرار کدها، بهبود کیفیت، افزایش کارایی و تسریع فرآیند توسعه نرم افزارها مؤثر است.

Django ۳-۲

جنگو، یک پلتفرم (یا چارچوب) توسعه برنامه‌های کاربردی در زبان پایتون محسوب می‌شود که برای ساختن و «نگهداری» (Maintaining) برنامه‌های تحت وب با کیفیت مورد استفاده قرار می‌گیرد. کتابخانه جنگو از طریق حذف وظایف تکراری، فرایند توسعه برنامه‌های کاربردی تحت وب را ساده و از لحاظ زمانی مقرون به صرفه می‌کند.

مزایای استفاده از Django که باعث شد در این پروژه از آن استفاده شود عبارتند از:

استقلال نسبی مؤلفه‌های مختلف این پلتفرم: جنگو به گونه‌ای طراحی شده است تا هر کدام از عناصر و مؤلفه‌های این پلتفرم، استقلال نسبی از یکدیگر داشته باشند.

کد نویسی کمتر: کد نویسی کمتر در جنگو، مترادف با چرخه سریع توسعه نرم‌افزار خواهد بود.

توسعه سریع: فلسفه طراحی جنگو به گونه‌ای است که به برنامه‌نویسان و توسعه‌دهندگان اجازه می‌دهد تا در کوتاه‌ترین زمان ممکن، برنامه‌های کاربردی را توسعه دهند.

طراحی تمیز: کدهای برنامه‌نویسی جنگو در پایتون بسیار خوانا و تمیز طراحی شده‌اند.

پشتیبانی از «نگاشت مدل اشیاء به مدل رابطه‌ای» (Object-Relational Mapping | ORM): پلتفرم Django، پلی میان «مدل داده» (Data Model) و «موتور پایگاه داده» (Database Engine) پدید می‌آورد و از مجموعه بزرگی از «سیستم‌های مدیریت پایگاه داده» (Database Management Systems) نظیر PostgreSQL، MySQL، و سایر موارد پشتیبانی می‌کند. همچنین، جنگو از پایگاه‌های داده NoSQL نیز از طریق انشعاب خاصی از این پلتفرم، به نام Django-nonrel پشتیبانی می‌کند.

«پشتیبانی چند زبانه» (Multilingual Support): پلتفرم جنگو، از طریق سیستم داخلی بین‌الملی‌سازی این پلتفرم، از توسعه وبسایت‌های چند زبانه پشتیبانی می‌کند.

پشتیبانی از پلتفرم‌ها و چارچوب‌های مختلف توسعه وب: پلتفرم Django، پشتیبانی داخلی و تعبیه شده از پلتفرم‌هایی نظیر Caching، RSS، Ajax و سایر موارد فراهم می‌آورد.

«واسط کاربری گرافیکی» (Graphical User Interface | GUI) مدیریتی: پلتفرم جنگو، یک واسط کاربری آماده به استفاده (ready-to-use) برای فعالیت‌های مدیریتی وبسایت و مؤلفه‌های آن در اختیار برنامه‌نویسان و توسعه‌دهندگان قرار می‌دهد.

۳-۳ Django Rest Framework (DRF)

یک فریمورک توسعه وب سرویس های RESTful است که بر پایه فریمورک محبوب جنگو ساخته شده است. با استفاده از این فریمورک، برنامه نویسان می توانند به سادگی و سرعت بالا وب سرویس های RESTful پیچیده را ایجاد، مدیریت و ارتباط برقرار کنند.

مزایای استفاده از Django که باعث شد در این پروژه از آن استفاده شود عبارتند از:

انتزاع و انعطاف پذیری: یکی از مزیت های بزرگ DRF این است که از مدل های Django به عنوان منبع اصلی برای داده ها استفاده می کند. این به برنامه نویسان این امکان را می دهد تا با استفاده از انعطاف پذیری مدل های Django سریع تر و با دقت بیشتری وب سرویس های RESTful را ایجاد کنند. هر تغییری که در مدل ها اعمال شود، به طور خودکار در وب سرویس ها نیز اعمال می شود.

مدیریت رابطه ها: مدل های Django می توانند روابط مختلفی را با یکدیگر داشته باشند (مانند روابط یک به چند و یا روابط چند به چند). DRF به برنامه نویسان این امکان را می دهد تا به سادگی این روابط پیچیده را در وب سرویس های خود نمایش دهند و از آن ها استفاده کنند.

اعتبارسنجی داده ها: یکی از ویژگی های مهم DRF این است که به برنامه نویسان اجازه می دهد تا داده ها را قبل از ذخیره سازی در پایگاه داده اعتبارسنجی کنند. این اعتبارسنجی می تواند شامل اعتبارسنجی انواع داده، اعتبارسنجی ارتباطات و روابط و یا هر گونه قانونی دیگری باشد که برنامه نویس می خواهد اعمال کند.

نمایش داده ها: DRF به برنامه نویسان امکان می دهد تا داده های مدل ها را در فرمت های مختلف مانند JSON یا XML نمایش دهند. این امکان به وب سرویس ها این امکان را می دهد تا با انواع دیگری از سیستم ها و اپلیکیشن ها ارتباط برقرار کنند.

به طور کلی، مدیریت کارآمد مدل ها در DRF به برنامه نویسان امکان می دهد تا با استفاده از مدل های Django، وب سرویس های RESTful پیچیده ای را با دقت و سرعت بالا ایجاد و مدیریت کنند. این ویژگی و امکانات جنگو بهره می برد و به برنامه نویسان این اطمینان را می دهد که داده ها به درستی مدیریت و نمایش داده می شوند.

۳-۴ djoser

جوسر Djoser یک کتابخانه drf است که مجموعه ای از کدهای آماده برای پیاده سازی اقدامات اساسی مانند ثبت نام، ورود به سیستم، خروج از سیستم، بازنشانی رمز عبور و فعالسازی حساب را فراهم می کند و صرفاً با مدل کاربر سفارشی شده در جنگو کار می کند.

۳-۵ JWT

مخفف JSON Web Token است و یک استاندارد باز و ایمن برای تبادل اطلاعات در قالب یک توکن است. این توکن ها به عنوان ابزاری برای احراز هویت و تأیید اعتبار کاربران در برنامه های وب و موبایل به کار می روند.

JWT بسیار کارآمد است و دلایل استفاده از آن در این پروژه عبارتند از:

برای احراز هویت کاربران و تأیید اعتبار آنها بکار می رود.

حجم کمی دارد و به راحتی قابل انتقال است.

استفاده از توکن های JWT به ازای هر درخواست، بار سرور را کاهش می دهد، زیرا نیازی به نگهداری وضعیت سمت سرور (مانند sessions) ندارد.

اطلاعات درون توکن امنیتی است و با استفاده از امضاء دیجیتال از تغییرات جلوگیری میکند.

۳-۶ Git

گیت (Git) یکی از محبوب ترین سیستم های کنترل ورژن (VSC) توزیع شده و متن باز جهان است که در سال ۲۰۰۵، لینوس توروالدز (Linus Torvalds)، خالق هسته سیستم عامل لینوکس، آن را ایجاد کرد. Git بیشتر برای ردیابی تغییرات فایل های پروژه به کار برده می شود. تعداد بسیار زیادی از پروژه های دنیا از سیستم گیت برای کنترل ورژن نرم افزارهای خود استفاده می کنند. همچنین، گیت روی بسیاری از سیستم عامل ها و IDE ها به طور پیش فرض وجود دارد که می توانید از آن استفاده کنید. هر نسخه از گیت، مخزنی شامل تاریخچه کامل تغییرات سورس مدنظر است. Git بسیار سریع و ایمن است که به توسعه پروژه های تیمی کوچک و بزرگ کمک می کند. گیت باتوجه به امکانات و ویژگی هایش، بهترین انتخاب برای اکثر تیم های نرم افزاری است. Git پایه بسیاری از سرویس ها مانند گیت هاب (GitHub) و گیت لب (GitLab) است.

۷-۳ VS Code

ویژوال استودیو کد یک کد ادیتور متن باز می‌باشد که با زبان‌های برنامه نویسی جاوا، جاوا اسکریپت، تایپ نوشته شده است. هرچند که نخستین نسخه‌ی ویژوال استودیو کد در سال ۲۰۱۵ CSS و HTML اسکریپت، منتشر شده است، اما سرعت رشد جامعه‌ی کاربران آن به قدری بالاست که امروزه به عنوان یکی از سه کد ادیتور برتر و پراستفاده‌ی جهان شناخته می‌شود. ویژوال استودیو کد در سیستم عامل‌های مختلف (ویندوز، مک‌اواس و گفته cross platform لینوکس) قابل نصب است؛ به همین دلیل به آن یک کد ادیتور چند سکویی یا می‌شود و یکی از علت‌های محبوبیت آن، همین است.

مزایای استفاده از VS Code که عبارتند از:

- حجم کم نرم‌افزار (زیر ۱۰۰ مگابایت) و سرعت بالای آن
- کاربرپسند بودن
- قابلیت ناوبری و جابه جایی سریع میان فایل‌ها، نمادها و کدها
- پشتیبانی از اغلب زبان‌های برنامه نویسی مانند جاوا، جاوا اسکریپت، Go، html، CSS، پایتون و...
- پشتیبانی از زبان‌های نشانه‌گذاری
- افزونه‌های عالی و کاربردی برای هر یک از زبان‌های برنامه نویسی
- تنظیمات اختصاصی و قابلیت‌های شخصی‌سازی گسترده برای هر پروژه
- کلیدهای میانبر بسیار کاربردی
- نصب پکیج برای استفاده از اسنپت‌های پیش فرض مخصوص هر زبان برنامه نویسی
- پشتیبانی از فناوری‌های گیت
- پشتیبانی عالی مایکروسافت از این کد ادیتور
- وجود یک جامعه‌ی گسترده‌ی فعال
- تغییر تم رنگی محیط ویرایشگر (همانند ویژوال استودیو)

۸-۳ Visual Paradigm

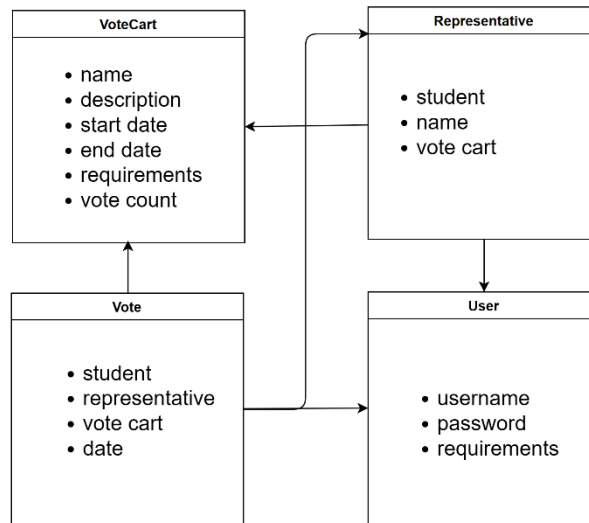
نرم افزار paradigm visual ، یک نرم افزار با محیطی جذاب و ساده جهت طراحی ERD است که در طراحی پایگاه داده به ما کمک می‌کند. این نرم افزار با استفاده از رابط کاربری ساده و ابزارهای آماده جهت رسم نمودار در روند ترسیم تسریع قابل توجهی می‌بخشد.

فصل چهارم: طراحی و ساختار

۴-۱ طراحی پایگاه داده

۴-۱-۱ مدل ERD سیستم

همانطور که در ERD قابل مشاهده است نیاز به ۴ جدول داریم که در ادامه این جداول را بررسی می کنیم.



شکل 1-4 شمای کلی پایگاه داده

۴-۱-۲ جدول User

جدول User وظیفه نگه داری از اطلاعات دانشجویان را برعهده دارد. همچنین این جدول از جدول پیشفرض جنگو برای مدل های کاربر ارث بری می کند و توانایی مدیریت را نیز می توان به دانشجویان داد. ویژگیهای موجود در این جدول در ادامه نشان داده شده است.

Attribute name	Type
id	integer
username	varchar
password	varchar
requirements	varchar

جدول 1-4 ویژگیهای جدول user

ستون اول id است که مقداری عددی بوده و primary key برای این جدول می باشد و استفاده اصلی آن برای پیدا کردن موجودیت در دیگر جدول ها است.

مقدار این ستون بصورت خودکار افزایش یافته و برای هر موجودیت منحصر به فرد می باشد.

هر دانشجو برای ورود به سیستم به شماره دانشجویی و رمز نیازمند است که بر اساس اطلاعاتی که از قبل و بصورت دستی به پایگاه داده وارد شده است، می تواند اقدام کند.

ستون مربوط به username برای شماره دانشجویی می باشد و ستون password نیز رمز دانشجو می باشد که طبق داده های وارد شده شماره ملی ثبت شده در سامانه گلستان دانشجو می باشد.

ستون requirements کاربر برای مشخص شدن رای گیری که کاربر می تواند در آن شرکت کند است.

از آنجایی که برای انواع دیگر رای گیری نیاز به اطلاعاتی محرمانه از دانشجو دارد و دانشگاه چنین اطلاعاتی را در اختیار ما نگذاشته است و همچنین نحوه آماده سازی پایگاه داده بصورت دستی و بر اساس اطلاعات پایه ای که دانشگاه در اختیار ما گذاشته است، نیاز به دیگر پارامتر ها جهت احراز هویت برای انواع مختلف رای گیری نمی باشیم.

این موجودیت تعداد زیاد دیگری از پارامتر ها را دارا می باشد که مربوط به مدل پیشفرض کاربر جنگو می باشد که ما از آن ارث بری کردیم که از آنجایی که مدل ما شخصی سازی شده است نیازی به توضیح تمامی این پارامتر ها نمی بینیم.

تنها پارامتری که در این مدل اهمیت دارد، برای تعیین سطح دسترسی دانشجویان است که از پنل مدیریتی پیشفرض جنگو و توسط ادمین اصلی ای که هنگام پیاده سازی پروژه تعیین شده است امکان پذیر می باشد.

۳-۱-۴ جدول representative

جدول representative وظیفه نگه داری از اطلاعات کاندید ها را برعهده دارد. ویژگیهای موجود در این جدول در ادامه نشان داده شده است.

Attribute name	Type
id	integer
student	foreignkey
name	varchar
vote_cart	foreignkey

جدول 2-4 ویژگیهای جدول representative

این جدول مختص به نمایندگانی است که در رای گیری شرکت کرده اند.

در ستون اول ایدی قرار دارد که مشابه با توضیحاتی که برای موجودیت قبلی دادیم عمل می کند.

در ستون student که یک foreignkey به جدول دانشجویان است، ما هر نماینده را نیز به عنوان دانشجو شناخته و اطلاعات اولیه ی نمایندگان را از این جدول و تحت عنوان دانشجوی معمولی که خود نیز حق رای دارد دریافت می کنیم.

در ستون name ما علاوه بر مشخصاتی که به عنوان دانشجو دریافت کردیم، نیاز به نام نماینده جهت نمایش در رای گیری داریم که در این فیلد ذخیره می کنیم.

و در آخر ستون vote_cart برای مشخص کردن رای گیری ای دانشجو در آن نماینده شده است تعبیه شده است.

۴-۱-۴ جدول votecart

جدول **votecart** وظیفه نگه داری از مشخصات رای گیری ساخته شده توسط ادمین را برعهده دارد. ویژگیهای موجود در این جدول در ادامه نشان داده شده است.

Attribute name	Type
id	integer
name	varchar
start_date	date
end_date	date
description	varcahr
requirements	varcahr

جدول 3-4 ویژگیهای جدول elections

هر رای گیری که توسط ادمین ساخته می شود نیاز به یک نام دارم و باید تاریخ شروع و پایان و همچنین توضیحاتی را نیز برای آن مشخص کند.

فیلد requirements که در جدول کاربران نیز وجود دارد، در این قسمت مقایسه می شود و در صورت برابر بودن دانشجو توانایی شرکت در این رای گیری را خواهد داشت.

۵-۱-۴ جدول votes

جدول vote وظیفه نگهداری از رای های داده شده به کاندیدها را برعهده دارد. ویژگیهای موجود در این جدول در ادامه نشان داده شده است.

Attribute name	Type
id	integer
student	foreignkey
representator	foreignkey
date	date
vote_cart	foreignkey

جدول 4-4 ویژگیهای جدول votes

هر کاربر با توجه به vote cart می تواند در رای گیری مربوطه که در vote code خود نیز مشخص شده است شرکت کند.

در ستون representator نماینده ای که vote cart ان نیز به این رای گیری اشاره می کند قرار دارد و دانشجو می تواند نماینده خود را انتخاب کند و به آن رای بدهد.

ستون date تاریخ لحظه ثبت رای را ذخیره می کند که دو استفاده از ان می شود. اول اینکه اگر دانشجو رای خود را تغییر داد ، رای با تاریخ قدیمی تر حذف می شود. همچنین، از این مقدار برای مقایسه با تاریخ شروع و پایان رای گیری استفاده می شود و در صورتی که تاریخ ثبت شده در این فیلد بین تاریخ شروع و پایان رای گیری بود، در جدول ذخیره می شود.

۴-۲ احراز هویت

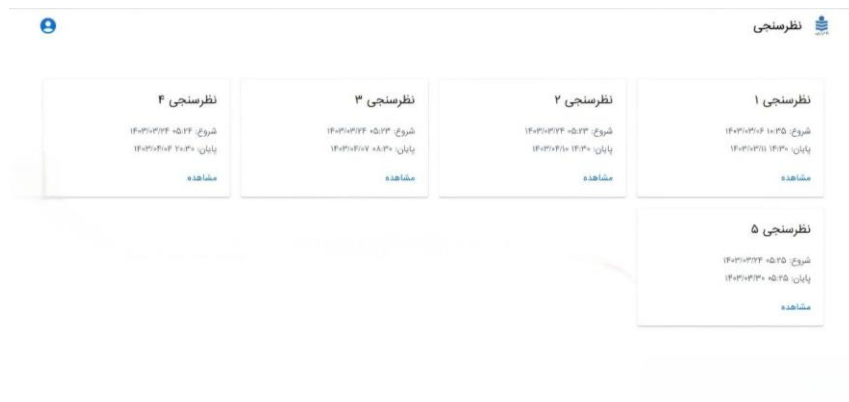
یکی از روشهای احراز هویت و اعتبار سنجی سمت سرور برای برنامه های سمت کاربر استفاده از Token-Based Authentication است که بر مبنای ارسال یک توکن امضا شده به سرور به ازای هر درخواست است. برای این کار از ترکیب JWT و Djoser استفاده میکنیم .

کاربر برای ارسال درخواستهای مربوط به ثبت نام و ورود نیاز به توکن ندارد ولی بعد از ثبت نام یا ورود دانشجو یا ادمین، یک توکن بر اساس شماره دانشجویی یا نام کاربری ادمین و در بخش پاسخ درخواست فرستاده می شود. در این پروژه همه اندپوینت ها بجز اندپوینت های ثبت نام و ورود محدودیت دسترسی دارند و نیاز دارند که کاربر توکن خود را در هدر مربوط به authorization ارسال کند. سرور به ازای هر درخواست ابتدا معتبر بودن توکن و تطابق نقش و دسترسی درخواست ارسال شده را بررسی میکند و اگر دسترسی کاربر به آن اندپوینت مجاز بود اطلاعات درخواست شده را از پایگاه داده استخراج کرده و ارسال میکند.

۴-۳ تصاویر نهایی



شکل 2-4



شکل 3-4



شکل 4-4



شکل 5-4

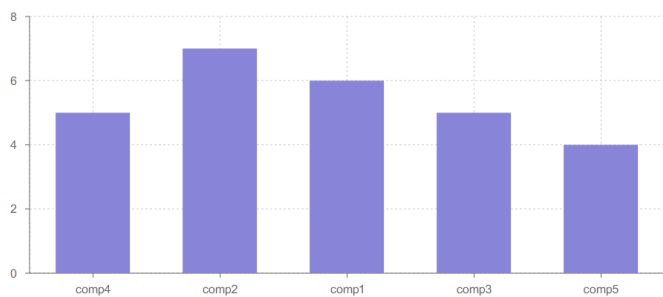


انتخابات انجمن علمی رشته کامپیوتر

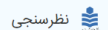
پایان: ۱۴۰۳/۰۴/۲۶ ۱۹:۰۰

شروع: ۱۴۰۳/۰۴/۲۵ ۰۷:۰۰

انتخابات انجمن علمی رشته کامپیوتر لطفا رای خود را ثبت نمایید.



شکل 4-6



- نظرسنجی ها
- دانشجوها
- نمایندگان



افزودن نماینده

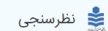
نمایندگان

انتخابات انجمن علمی رشته کامپیوتر

فیلتر بر اساس نظرسنجی:

نام کاربری	نام	نظرسنجی	تعداد رای	اقدامات
comp1	1	انتخابات انجمن علمی رشته کامپیوتر	۶	 
comp2	2	انتخابات انجمن علمی رشته کامپیوتر	۷	 
comp3	3	انتخابات انجمن علمی رشته کامپیوتر	۵	 
comp4	4	انتخابات انجمن علمی رشته کامپیوتر	۵	 

شکل 4-7



- نظرسنجی ها
- دانشجوها
- نمایندگان

فصل پنجم: پیاده سازی

۵-۱ مقدمه

در این بخش به نحوه پیاده سازی اندپوینت‌های بکند، ورودی و خروجی‌های مورد نیاز آنها می‌پردازیم.

۵-۲ کنترلر ادمین

در این بخش به توضیح اندپوینت‌های مربوط به ادمین می‌پردازیم. ادمین‌ها پس از ثبت نام یا ورود با استفاده از توکنی که برای آنها ساخته می‌شود می‌توانند به همه اندپوینت‌های مربوط به خود دسترسی پیدا کنند. در این مرحله از پروژه اولین ادمین بصورت پیش فرض ساخته می‌شود و اطلاعات حساب کاربری آن در اختیار فرد مسئول قرار می‌گیرد و این فرد دسترسی دارد که ادمین‌های دیگری به برنامه اضافه کند.

۵-۲-۱ ورود

این اندپوینت زمانی استفاده می‌شود که هر کاربری از جمله ادمین در برنامه حساب کاربری داشته باشد ولی توکن معتبری ندارد که می‌تواند بدلیل خروج از حساب کاربری، نیاز به ورود از طریق دستگاه یا مرورگر دیگر یا منقضی شدن توکن باشد.

ادمین برای ورود نیاز به ارسال نام کاربری و رمز عبور دارد.

Method	Post
Url	"/login"

جدول 1-5 اطلاعات اندپوینت ورود ادمین

۵-۲-۲ ثبت نام و لیست ادمین‌ها

در این درخواست باید فیلد مربوط به ادمین شدن کاربر مورد نظر توسط ادمین پیش فرض فعال شود.

Method	Post
Url	"/admin/core/user/"

جدول 2-5 اطلاعات اندپوینت ثبت نام ادمین

۳-۲-۵ پروفایل و ویرایش اطلاعات ادمین

با استفاده از این اندپوینت می توان اطلاعات حساب کاربری یک ادمین را بدست آورد.

Method	Get
Url	"/admin/core/user/userId/change/"

جدول 3-5 اطلاعات اندپوینت پروفایل ادمین

۳-۵ کنترلر دانشجو

در این بخش به توضیح اندپوینت های مربوط به دانشجویان میپردازیم. دانشجویان پس از ثبت نام یا ورود با استفاده از توکنی که برای آنها ساخته می شود می توانند به همه این اندپوینت ها دسترسی پیدا کنند.

۱-۳-۵ ثبت نام

در این درخواست باید اطلاعات دانشجو مانند شماره دانشجویی و رمز عبور و نیازمندی دانشجو ارسال شود، در نتیجه یک سطر جدید در جدول user برای این دانشجو ساخته می شود.

Method	Post
Url	"/admin/students/"

جدول 4-5 اطلاعات اندپوینت ثبت نام دانشجو

۲-۳-۵ ورود

این اندپوینت زمانی استفاده می شود که دانشجو در برنامه حساب کاربری داشته باشد ولی توکن معتبری ندارد که می تواند بدلیل خروج از حساب کاربری، نیاز به ورود از طریق دستگاه یا مرورگر دیگر یا منقضی شدن توکن باشد. دانشجو برای ورود نیاز به ارسال شماره دانشجویی و رمز عبور دارد.

Method	Post
Url	"/login"

جدول 5-5 اطلاعات اندپوینت ورود دانشجو

۳-۳-۵ پروفایل

با استفاده از این اندپوینت می توان اطلاعات حساب کاربری یک دانشجو را بدست آورد. این اندپوینت صرفاً برای ادمین ها قابل مشاهده است.

خود کاربر نیز در تمامی صفحه ها، از آیکون پروفایل می تواند اطلاعات کاربری خود که تنها شامل نام کاربری است را مشاهده کند.

Method	Get
Url	"/admin/students/"

جدول 5-6 اطلاعات اندپوینت پروفایل دانشجو

۴-۵ کنترلر کاندید

در این بخش به توضیح اندپوینت های مربوط به نامزدهای انتخابات می پردازیم. اندپوینت های این بخش تنها در دسترس ادمین هاست و باقی کاربران و دانشجویان به آنها دسترسی ندارند. در هدر این درخواست ها نیز باید توکن معتبر مربوط به ادمین ارسال شود تا سرور اجازه دسترسی به اطلاعات را صادر کند.

۴-۵-۱ ثبت نام، ویرایش و لیست همه نامزدها

در این درخواست باید اطلاعات یک نامزد انتخاباتی که شامل نام و رای گیری ای که در آن کاندید شده است و کاربری که تقاضای نماینده شدن را دارد ارسال شود. به ازای هر فرد یک سطر جدید در جدول representative اضافه می شود.

Method	Post
Url	"/admin/representatives"

جدول 5-7 اطلاعات اندپوینت ثبت نام نامزد انتخابات

۵-۵ کنترلر انتخابات

در این بخش به توضیح اندپوینت‌های مربوط به انتخابات می‌پردازیم. برخی از این اندپوینت‌ها تنها در دسترس ادمین‌ها و برخی دیگر در دسترس دانشجویان است. در هدر این درخواست‌ها نیز باید توکن معتبر مربوط به ادمین یا دانشجو ارسال شود تا سرور اجازه دسترسی به اطلاعات را صادر کند.

۵-۵-۱ ثبت انتخابات ، لیست همه انتخابات‌ها

در این درخواست باید اطلاعات یک انتخابات که شامل نام، تاریخ شروع و پایان، توضیحات، تعداد مجاز رای هر دانشجو می‌باشد ارسال شود. یک ویژگی نیازمندی‌ها نیز در جدول VoteCart وجود دارد که می‌توان شروطی را برای شرکت در انتخابات مشخص کرد تا فقط گروه خاصی از دانشجویان اجازه شرکت در آن انتخابات را داشته باشند. این اندپوینت تنها در دسترس ادمین‌هاست و دیگر کاربران اجازه ساخت انتخابات جدید را ندارند.

Method	Post
Url	"/admin"

جدول 5-8 اطلاعات اندپوینت ثبت انتخابات

۵-۵-۲ نتیجه یک انتخابات برای ادمین

این درخواست نیز نیاز به دسترسی ادمین دارد و شناسه یک انتخابات را دریافت می‌کند و در پاسخ اطلاعات انتخابات و نامزدهای آن انتخابات به همراه تعداد رای‌های آنها را برمی‌گرداند.

Method	Get
Url	"/admin/vote_carts/{voteCartId}"

جدول 5-9 اطلاعات اندپوینت نتیجه یک انتخابات برای ادمین

۳-۵-۵ لیست همه انتخابات‌های دانشجو

این درخواست مربوط به دانشجو است و هیچ اطلاعاتی دریافت نمی‌کند. درواقع شماره دانشجویی این فرد را از توکن ارسالی در هدر درخواست استخراج کرده و در پاسخ لیست انتخابات‌هایی که این دانشجو امکان شرکت در آنها را دارد را ارسال می‌کند.

Method	Get
Url	"/dashboard"

جدول 5-10 اطلاعات اندپوینت لیست همه انتخابات‌های دانشجو

۴-۵-۵ لیست نامزدهای یک انتخابات

این درخواست مربوط به دانشجو است. شناسه یک انتخابات را می‌گیرد و در جواب لیستی از نامزدهای آن انتخابات را ارسال می‌کند.

Method	Get
Url	"/dashboard/vote_carts/{voteCartId}"

جدول 5-11 اطلاعات اندپوینت لیست نامزدهای یک انتخابات

۵-۶ کنترلر رای

در این بخش به توضیح اندپوینت مربوط به ثبت رای میپردازیم. این اندپوینت در دسترس دانشجویان است. در هدر این درخواست نیز باید توکن معتبر مربوط دانشجوی ارسال شود تا سرور اجازه دسترسی به اطلاعات را صادر کند.

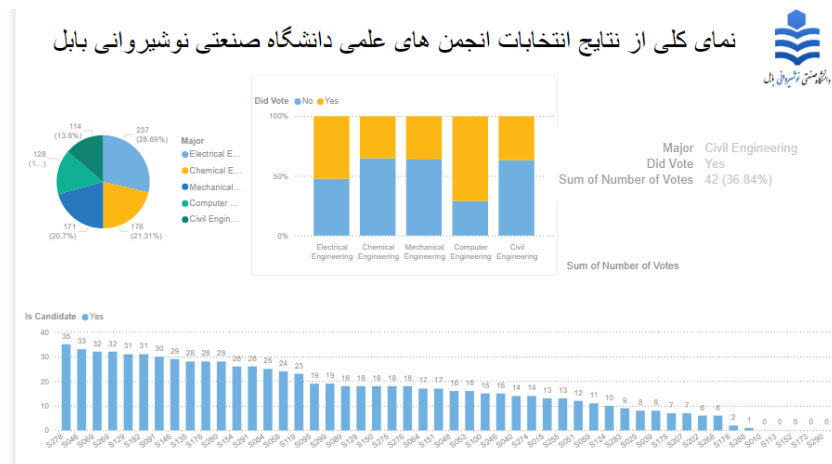
۵-۶-۱ ثبت رای

در این درخواست باید اطلاعات یک رای که شامل شناسه انتخابات و شناسه نامزد می باشد ارسال شود. بعد از ارسال این درخواست ابتدا بررسی می شود که این دانشجو قبلا در این انتخابات شرکت نکرده باشد، در غیر اینصورت پیغام خطایی با عنوان رای تکراری در پاسخ درخواست ارسال می شود. اگر دانشجو برای اولین بار در این انتخابات شرکت کرده باشد، یک واحد به تعداد رایهای نامزد مورد نظر و همچنین یک سطر به جدول vote اضافه می شود و شناسه انتخابات و شناسه دانشجویی که رای داده است و نماینده ای که به آن رای داده است در آن ذخیره می شود.

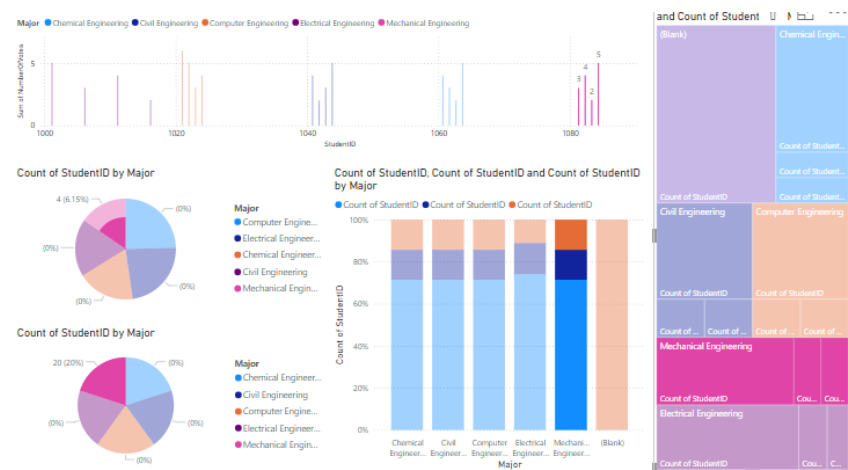
Method	Post
Url	"/dashboard/vote_carts/{voteCartId}"

جدول 5-12 اطلاعات اندپوینت ثبت رای

۵-۷ خروجی نهایی دیتابیس در نرم افزار power bi



شکل 5-1



شکل 5-2

فصل ششم: نتیجه گیری و پیشنهادهای

پروژه رای گیری انجمن های دانشجویی به رفع نیاز به یک سیستم مدرن و کارآمد برای تسهیل انتخابات دانشجویی و ترویج مشارکت دموکراتیک در دانشگاه پرداخته است. در این پروژه از طریق پیاده سازی یک پلتفرم مبتنی بر وب، فرآیند رای گیری را ساده سازی کرده ایم، کاغذبازیهای دستی را حذف کرده ایم و برای دانشجویان یک رابط کاربری در دسترس و کاربرپسند برای دادن رای خود فراهم کرده ایم.

این پروژه به اهداف اولیه خود یعنی افزایش شفافیت، امنیت و راحتی انتخابات دانشجویی دست یافته است. با استفاده از قدرت فناوری های وب، ما دانشجویان را برای مشارکت فعال در فرآیند دموکراتیک توانمند کرده ایم و از یکپارچگی نتایج رای گیری اطمینان حاصل کرده ایم. اتخاذ اصول طراحی واکنشگرا، تجربه ثابتی را در دستگاه های مختلف تضمین می کند و به دانشجویان اجازه می دهد تا از پلتفرم های مورد علاقه خود در انتخابات شرکت کنند. همانطور که به جلو می رویم، چندین راه برای تقویت و گسترش پروژه رای گیری انجمن های دانشجویی وجود دارد تا ارزش بیشتری برای دانشگاه ما ایجاد کند:

پشتیبانی از نقشه ای مختلف: برنامه را برای پشتیبانی از نقشه های مختلف مانند مدیران انتخابات، نامزدها، ناظران و رای دهندگان با سطوح دسترسی و داشبورد مناسب گسترش دهیم.

بررسی های امنیتی: انجام بررسی های امنیتی دوره های برای اطمینان از استحکام مکانیسم های احراز هویت و مدیریت داده ها.

دسترس پذیری: ویژگی های دسترس پذیری برنامه را افزایش دهیم تا اطمینان حاصل کنیم که برنامه برای افراد دارای معلولیت نیز به راحتی قابل استفاده است.

گرفتن بازخورد: مکانیسمی برای ارائه بازخورد توسط رای دهندگان و کاربران برنامه ایجاد شود تا بتوان زمینه هایی برای بهبود و اصلاح برنامه شناسایی کرد.

- [1] [Back End و Front End](#)
- [2] [پایگاه داده چیست و چه کاربردی دارد؟](#)
- [3] [What Is An API \(Application Programming Interface\)?](#)
- [4] [API Endpoints - What Are They? Why Do They Matter?](#)
- [5] [Backend Controllers](#)
- [6] [What is a Programming Library?](#)
- [7] [What is a Framework in Programming?](#)