

Online Advisement Platform:



Senior Design I/II Final Proposal

Chrystal Mingo, Rehman Arshad, Xunshan Lin, and Zhicong Wen

I. PROJECT CONTEXT

THE goal of this project is to create an automated advising platform called AdviseMe. That allows students at the City College of New York (CCNY) to track their academic progress and automate advisement. Students will complete a live advisement form and get notifications through our workflow system when they are eligible for enrollment. AdviseMe will provide a more user-friendly and efficient advising experience for both students and advisors to raise the graduation rate at CCNY and help us transition to virtual advisement due to the pandemic.

A. Problem Description

Advisement is a prerequisite that CCNY college students must face every semester. Due to the Covid-19 pandemic, many students are forced to do advising online. This is an issue for both advisors and students, as any workflow being relegated to emails is inefficient and bound to have failures.

The problem many colleges face is that they have a low graduation rate. For instance, in 2018, students seeking a bachelor's degree at CCNY completed within 4 years were only 16.2%, In 5 years 46.9%, and 6 years 55.3%. The graduation rate gives students insight into how many students are finishing their degrees promptly once they enroll. One of the main reasons for low graduation rates is the lack of academic preparation, such as not having enough resources to inform students what courses they need to take.

One of the success keys for graduating from college is creating a four-year degree plan. Unfortunately, there is only one advisement website that students have access to at CCNY. However, it only serves as an academic tracker, and it does not automate advisement. As a result, CCNY advisement is dependent on an email workflow. The lack of academic preparation mainly delays the completion of a degree. To avoid spending plenty of time designing a long-term degree plan individually, a user-friendly advising website can help students stay on track and increase graduation rates properly.

B. Previous Work

Although our idea of the project is unique, there are some solutions to help students design personal academic plans. Firstly, an example that currently exists on CUNYfirst is

called "DegreeWorks", which we have drawn inspiration from. "DegreeWorks" is a web-based planning tool to help students and advisors monitor students' progress toward degree completion. It includes an overview of the student, a GPA calculator, a list of courses needed to graduate, and a degree path.

While "DegreeWorks" has plenty of core functionalities, many functionalities are sometimes not user-friendly and well-organized. The way courses are displayed to graduate is not intuitive, and it can be overwhelming to look at, occasionally unorganized, and even displays the wrong information. Furthermore, the student's overview in "DegreeWorks" is a small area towards the top, showing only the credits. We hope to replace "DegreeWorks" at CCNY with our automated online advisement platform called "AdviseMe".

Similarly, some online advising platforms currently exist in the market are Target X and Skyward Student Management that are not user friendly. Unfortunately, due to COVID-19, the in-person advising process has become strained. However, there is no CUNY-based advising platform that exists to help CCNY students complete advisement online. We plan to combine some of the core functionalities of "DegreeWorks" and implement our live advisement form to automate advisement at CCNY as a comprehensive web application for CCNY students.

C. Our Solution

Our solution is creating a web application that serves as an improved and upgraded version of the "DegreeWorks" at CCNY. Our online advising platform is called "AdviseMe". We plan to make it more user-friendly than "DegreeWorks", it will help students create their personalized degree plans and, most importantly automate advisement at CCNY.

Once the student declares their major, our website will have all the functionality to generate a plan depending on your major and customize depending on your goals and roadblocks to finish college promptly. By providing a high-quality experience, we will build on the best parts of previous efforts and add a new core functionality to improve advisement at CCNY. Students will be able to fill out a live advisement form and receive feedback from their advisors on our platform. As well, they will see the status of their

advisement and be notified when eligible for enrollment.

AdviseMe will serve as a one-stop-shop for tracking degree completion, creating a personalized degree plan, and simplifying the advisement process for both students and advisors at CCNY.

II. PROJECT DESCRIPTION

The first updated functionality that we draw inspiration from "DegreeWorks" is a new creative interface for the student's profile. As users sign in to our platform, students' profile will display their basic information with details and an overview of their checklist. Once they have completed all their course requirements, students can apply for graduation with the click of a button. Students will also have direct-links to a personalized degree planner and our live advisement form.

This project's principal functionality is an automated live advisement form that allows students and advisors to interactively view completed courses and confirm intended courses for the following semester. This form will automatically fill in all the courses and grades needed to apply to the major or minor, including the classes that the student is currently enrolled in. As users finalize the expected courses for next semester, their grades are incorporated into the final advisement form and directly submitted to the advisor. This way, the advisor can directly check all the students in need of advisement, view their form, and add advising notes to improve their academic plan before accepting or declining an advisement form. Furthermore, a GPA calculator will display the student's current GPA based on their grades thus far. The student can also use the GPA calculator to estimate their GPA for the current semester.

A workflow system for students and advisors is crucial for live advising. The workflow system serves as an status bar to inform students when they are eligible for enrollment. Moreover, the workflow system is customizable, and students can filter and choose different visualization styles based on their viewing preferences.

A. Data ingestion

To implement our core functionalities, our project requires student/staff data and course data from CCNY. Since we can't gain access to this data from "DegreeWorks", we plan to simulate student/staff data by creating a registration form that would collect the necessary data needed for our use cases. We will generate dummy data to fill in the student/staff basic information and student's grades. Course data and degree requirements can be found on the CCNY website, and we can directly obtain the data required from each department's curriculum.

B. Application Prototype:

We have created a prototype of the core functionalities of our online advisement platform using Figma. Figma serves

as a visual of our application interface. Figma is an excellent platform for website design because "Figma to HTML" is a Figma plugin that helps convert your Figma design to code easily. The platform has two different views depending on if the user is a student or staff at CCNY.

The first version of the platform we will delve into is the student version.

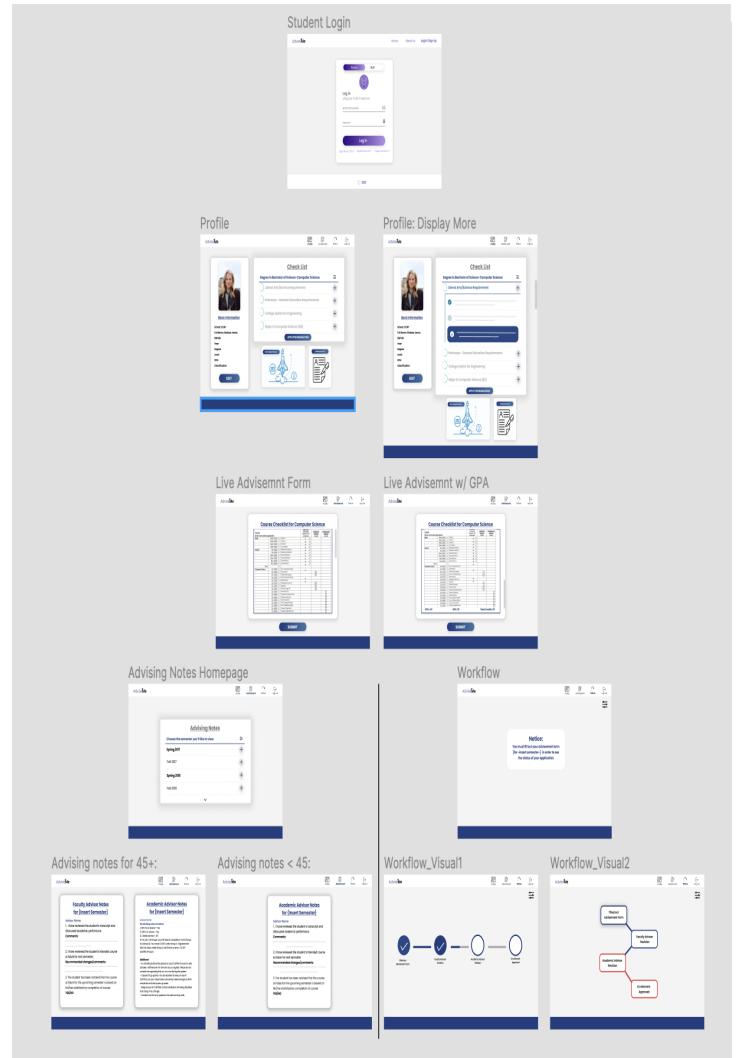


Figure 1: Student UX/UI Overview

Figure 1 serves as an overview of the student view prototype. On this platform, students will log in using their CCNY credentials. Upon logging in, they will be directed to the homepage, referred to as "Profile". This page will have their basic information, displays the course requirements, and track their degree progression, and it has direct links to a personalized degree planner and our live advisement form. Students can also view their advising notes and submit their live advisement directly from AdviseMe. And lastly, they have a workflow with two different visuals for the student to see the process of their advisement and are alerted once they are eligible for enrollment.

The second version of the platform we will delve into is the staff/advisor version.

Figure 2: Advisor UX/UI Overview

Figure 2 serves as an overview of the staff version of the prototype. On this platform, staff will log in using their CCNY credentials. Upon logging in, they meet with the main page, which will have basic information about the staff and showcase the list of students in need of advisement. Once the advisor selects a student, they will see the advisement form, give the student advising notes, and approve or reject the form. Staff also has the power to edit the live advisement form, as well as the workflow permissions needed to get approved for enrollment.

For more information and a closer look at the UX/UI of the student and advisor view, go to Appendix A: Application Prototype.

C. Discussions on Design Choice:

Our goal is to recreate and upgrade DegreeWorks, which serves as the current online advisement platform, and to automate advisement at CCNY. DegreeWorks is a web-based program that enables you to track your academic progress easily. Our platform AdviseMe not only allows students to track their academic progress but also automates advisement. At CCNY, students are required to fill out their advisement

form and send it via email to their advisors due to the pandemic. Yet, we have DegreeWorks at CCNY, and it is barely used for its primary purpose, and sometimes the information is incorrect. DegreeWorks also repeats certain courses on their list. We plan to make our requirement checklist concise and only display the necessary information.

Therefore, we created AdviseMe. Students will use this platform to track their academic progress and quickly fill out their advisement form with no worries of missed emails.

For our platform, we made some design choices keeping our users in mind regarding what they felt DegreeWorks lacked. Our Main Page for students encompasses the primary function of DegreeWorks. However, it is much more user friendly.

Student ID	Name	Degree	Major	Level	Classification	Last Audit
23643954	Mingo, Chrystal	BS	Computer Science (BS)	U	U-SENR (4)	12/17/2020

Figure 4: DegreeWorks Student Overview

Figure 5: DegreeWorks Requirements Overview

versus.

Figure 6: AdviseMe Main Page

Compared to DegreeWorks, instead of a small overview of the student, we allow our users to upload their basic information; Such that the advisor knows who they're advising, their goals, their desired job, etc. Students are allowed to edit their basic info and add their bio or resume to the platform. While DegreeWorks only has a small bar at the top, giving minor details about the student. Also, DegreeWorks can be very overwhelming because it lists out all the required classes

needed to graduate all at once. While on our platform, students will be able to filter their academic courses to see what they still need to take, courses that are already taken, and by default, each section will display all the required courses under that tab. Overall, AdviseMe will be a much more user-friendly platform and will be used to automate advisement at CCNY.

III. TECHNOLOGY DESCRIPTION

A. Overview

When considering which technologies to use during development, the following considerations were made:

- Ease of implementation
- Scale-ability and code re-usability
- Framework with light-weight web development libraries and robust tool-kits

After much discussion and analysis while also considering each team members strengths, weaknesses, and skill sets brought to the table; From all the various different web development technologies and frameworks, we finalized on selecting a tech stack that included:

- HTML5
- CSS3
- Python3
- Flask
- SQLite3
- Heroku

B. HTML5 and CSS3

HTML5 (Hypertext Markup Language), and CSS3 (Cascading Style Sheets) are the basic component of any web application. What makes them special in Python's flask framework is the Jinja2 the templating Syntax engine which completely redefines how HTML and CSS can be utilized. By using Jinja2 syntax we can pass Python variables and grammar into our web pages allowing us to display data, check conditionals statements, and display elements inside of loops.

C. Python

Python is an interpreted, object-oriented, and high-level, general-purpose programming language. With its concise yet robust syntax, a plethora of resources, precise documentation, and community packages and tools, Python has become quite popular. From data science to GUI programming, to machine learning, to web development, Python has a myriad of libraries that make development for any subject area more straightforward and time-efficient. The following Python libraries offers a wide variety of useful modules such as:

- OS
- Date-util
- Pandas
- NumPy
- Cryptography

Using Python's simple syntax and the date and time libraries we can build a modular, scalable, and robust web application that serves our purpose.

D. Flask

Flask is a lightweight Web Server Gateway Interface (WSGI) web application framework that allows for fast, efficient web development. Flask specializes in customize-ability. There is no default model in Flask. Which means one can add the necessary packages to make their application fall under Model View Controller (MVC) architecture or Model View Template (MVT) architecture (typically using some Object Relational Mapper (ORM) like SQLAlchemy). Flask is primarily suited for web development and can be equipped with useful extension packages such as:

- Flask-wtf
- Flask-login
- Flask-sqlAlchemy (ORM)
- Flask-itsdangerous
- Flask-censor

E. SQLite3

SQLite is a relational database management system contained in a C library. In contrast to many other database management systems, SQLite is not a client-server database engine. Rather, it is embedded into the end program. SQLite generally follows PostgreSQL syntax. The biggest difference between a normal Relational Database Management System (RDMS) like PostgreSQL is that SQLite stores the changes on a local file, which makes this volatile storage which is why it is not ideal for production environments. However, SQLite is a lightweight database that is perfect for testing out an MVP.

F. Heroku

Originally we planned to use AWS but Heroku seemed like a better option due to it's ease of deployment, and automated crash logging making debugging issues during deployment and overall DevOps much more simpler. Its common knowledge Heroku is built on top of AWS. Meaning we could eventually re-deploy the application to AWS or another cloud service provider. However SQLite3 is not recommended for a Heroku deployment, given it's not a fully fledged RDMS, like PostgreSQL and is considerably in a more volatile state if deployed given Heroku wipes all data stored on memory, RAM, cache, and local storage on all servers every 24 hours. Some of the things we considered

IV. DESIGN PHASE CONCLUSION

The ongoing process of advisement at The City College of New York is during an ongoing pandemic can only be described as confusing, inefficient, time consuming, and bound to fail whenever a chain of emails gets lost in all the inevitable wave of SPAM and phishing emails that Citymail users are often subjected to. Manually filling out the same form every semester is at best inefficient and at worse confusing, because students can make mistakes. Such as confusing their free electives and liberal arts. This is especially more common for Rising Juniors and Seniors as they are in the process of completing their core Computer Science courses in the curriculum, and they barely remember which course they took

3-4 years or 6-8 semesters ago. The typical reason for these errors often lies because these students don't use *Degree Works*, because it's difficult to use, and not user friendly. As a result we created AdviseMe an online advisement platform, that will take away the hassle that comes with advisement process at CCNY.

V. ENGINEERING PHASE

A. Implementation

Originally when developing *adviseme* our proposed technologies included, Python, Flask, SQLAlchemy as the (ORM), PostgreSQL as our Relational Database Management System (RDMS), and AWS Light-Sail for deployment. While the final product still heavily incorporated all of the core technologies we proposed, we were unable to deploy the application to AWS Light-Sail due to running into technical issues with PostgreSQL. We added SQLite as our database to address some of the technical issues we faced during development. We picked flask, because everyone on the team was familiar with Python and Flask had amazing libraries such as Flask-WTF (WT Forms) that we knew would be useful for creating:

- Live Advisement Form
- Curriculum Edit Form
- GPA and QPA Calculator
- Editable Workflow
- Upload and View Media
- Export Graduation List Excel, CSV spreadsheets

And also we were quickly able to identify tools and library documentation that helped us get started with a fully fledged User Authentication System using the Flask-Login library. We took full advantage of our ORM SQLAlchemy, by creating various tables with many-to-one, one-to-many, and many-to-many relationships. Users would have 2 other tables, with a foreign key relationship with the Faculty Table, and another foreign key relation with the Student Table. The Faculty table had a one-to-many relationship with the Notes Table, and the Student Table had a many-to-many relationship with the Courses Table, with the Enrollment Table being the association table between the two. The Enrollment Table also stored all the grades for the students in this Table.

B. Live Advisement Form

The live advisement form handles adding student courses to the currently enrolled or intend to take lists. It also has a form that accepts the semester and the students transcript. The Advisement Form cannot be submitted to any advisor (Faculty or Academic) without uploading the transcript as a PDF file. Only PDF files are accepted. On the Faculty side a whenever a student is below 45 credits their advisement form is sent to an academic advisor for approval. However, if the student is above 45 credits then their advisement form needs approval from both academic and faculty advisors. In the event a student's advisement form is rejected they can view the advisors notes, while sending an amended advisement form.

C. Curriculum Edit Form

For the faculty curriculum edit form, we used a SelectField wrapped inside of a FieldList, which are libraries from flask-wtf. Using this we implemented a form that allowed us to edit all the course data simultaneously. The issue is that unlike asynchronous JavaScript, and frameworks like React.js with its Virtual DOM manipulation, our flask application by comparison is very static and making it dynamic is extremely difficult given the limitations of the framework and WSGI model itself. Because the WSGI model relies on what is typically known as a GET, POST redirect every time it needs to update anything in a template no matter how small or trivial.

We needed to make a clever workaround from the static WSGI model that Flask is built on top of because it would not correctly update the fields needed and run into numerous issues. Meaning anytime a course was updated there were no issues, but if a course was added or deleted from the database it would cause issues. Our workaround although not optimal was to count the courses in Table an an optimized id based query, and then in a not so optimal manner append the fields so we always have correct amount. This workaround is stable and works but it could be more optimal.

D. Editable Workflow

The workflow determines whether students below 45 credits only see academic advisors, and students above 45 credits need advising from both academic and faculty advisors. This is the default setting, however the editable workflow enables faculty users in *adviseme* to actually change the workflow of how students get advised. They can preview their changes before making them. They can require both students below 45 credits and above 45 credits are mandated to get advising done from both academic advisors and faculty advisors. This gives *adviseme* a lot more flexibility and adaptability if the rules ever change in the future, as the platform has a modular workflow system for advisement.

E. GPA and QPA Calculator

The GPA and QPA calculator function calculated any students GPA and QPA based on the following formula. The course grade would return a numerical point value for each grade ranging from A-F, this would be multiplied by the course credits, and then finally this value would be divided by the students total credits they have earned overall meaning this was not a semester GPA, but a cumulative GPA calculation. In future revisions adding a semester GPA calculation would be the next goal. The QPA calculator function works slightly differently. The course grade would return a numerical point value for each grade ranging from A-F. However here the difference here is that the numerical point values for calculating a QPA is entirely different so a grade of an A would return a +2, a B would return a +1, a C would return a 0, a D would return a -1, and an F would return a -2. Another difference is that this function would only execute if a course is a CS course with the id range of 1-38. Code can be found here: https://github.com/zwen000/SDG6/blob/main/clean_up.py <https://github.com/zwen000/SDG6/blob/main/adviseme/routes.py>

F. Upload and View Media

We implemented a method to save profile images for the users. Rather than using any library we just used a form to accept the uploaded file, only accepting ".png", and ".jpg" files and then we split the file name with it's file extension, create a random generated hash and then renamed the file with the new hash concatenated with the file extension. This was to ensure files of the same name would not exist and be accidentally overwritten. We also used this opportunity to optimize the profile images being uploaded by resizing them using the pillow library to compress the image size so that it would take less server space once deployed and in production. Code can be found on lines (92-108) <https://github.com/zwen000/SDG6/blob/main/adviseme/routes.py>

Using the same logic we created a save transcript function that would store a student user's uploaded transcript, only accepting PDF files, then it would also get the file extension from the file using the OS module in Python. From here it was a matter of naming the file appropriately. We decided to name the transcript file in the following format "EMPLID_Semester_random_hex". Concatenating the random hex at the end was to ensure that a student would not overwrite their old transcript files; We could have the older files stored in the archive for reference later. Code can be found on lines (110-121)

<https://github.com/zwen000/SDG6/blob/main/adviseme/routes.py>

Although not a function, another feature worth mentioning was how we were able to embed a PDF file onto an HTML template page allowing us to view the students transcript and the CCNY Computer Science Curriculum. A combination of Jinja2 templating syntax along with a deep understanding of the relative file location of the resource within the server and some general intuition of the structure of our app allowed us to display previously uploaded and stored PDF files inside an object tag in HTML. Now this object tag also brings its own responsive design challenges to the table, so more research will be needed here. Code can be found here on line 11:

https://github.com/zwen000/SDG6/blob/main/adviseme/templates/Transcript_Cirriculum.html

G. Export Graduation List Excel, CSV spreadsheets

Understanding the utility of Pythons amazing data analysis tools suite, and leveraging the amazing Pandas and CSV Libraries we implemented a function to read a database query, use csv writer to write the query into an empty .csv file, and then used Pandas to convert the .csv file into an excel.xlsx spreadsheet file finally saving both files into the specified file path directory in the server. Then we created a button that would allow users to download all the exported data as either a .csv or excel .xlsx spreadsheet file that contains all the students who are graduating, with their first name, last name, EMPLID, GPA, QPA, and Citymail email address. Code can be found on lines (976 - 997)

<https://github.com/zwen000/SDG6/blob/main/adviseme/routes.py>

VI. RESULTS:

A. Overview:

We are here with a purpose, to make online advisement easier for both students and faculty. In hopes of combating the low graduation and college drop out rates. As a result we created AdviseMe, which serves as a one-stop-shop for tracking degree completion, creating a personalized degree plan, and simplifying the advisement process for both students and advisors at CCNY. Below will be a brief overview of AdviseMe's core functionalities depending on the user. Our platform is broken up into two parts, which is the Student and Faculty Version. You can view the fully rendered and implemented pages we created for both user experiences in Appendix C: AdviseMe Implemented Frontend.

B. Non - Users Homepage:

All users are welcomed to our platform, upon clicking on AdviseMe they are presented with an animated "Welcome to AdviseMe" page. The homepage serves as an overview of our online advisement platform. On the left side you can see the features available for students, and on the right side the features for faculty are displayed, as well as our purpose at the end of the page. In addition, we created an about us page, so users can meet the developers and learn about our roles. We also made a fully functional login and register page. View fully rendered Non-User Homepage in Appendix C: AdviseMe Implemented Frontend → Non - Users Homepage Frontend.

C. Student Version:

The Student Version of AdviseMe, has four major features that we implemented:

1) Profile of Student:

This is where CCNY students are able to view their profile which has basic information such as: Email, Grade Level, EMPLID, Credits Completed, Credits In Progress, and their GPA (which is automatically calculated by AdviseMe). In addition to the basic information we also display the student's profile image, brief bio, and degree they are currently in the works of completing.

2) Checklist:

This page allows students to track their degree completion. It is broken up into course sections such as: Comp Sci Required Major Courses, CSC Electives, Technical Electives, Free Electives, and of course Math, Science, and Liberal Arts. They each have their own progress tracker to help students see their progression in each course section. Once a student has completed their entire checklist aka completed their degree, our application alerts the student with an apply for graduation button.

3) Live Advisement Form:

The live advisement form is one the biggest features we implemented since it automates advisement at CCNY, and makes the process to fill out what a student intends to take so much easier. This page has the students grades, and courses they're currently enrolled in checked off, and the student only has to check off the courses they intend to take for the incoming semester. Our platform, makes the process of filling out your live advisement form so much easier, and takes away the hassle of printing the CCNY advisement, manually filling in your grades, courses currently enrolled in and courses you intend to take, scanning and emailing your advisor(s) through email. On our platform we made advisement automated. Also students can view their advising notes from previous semester.

4) Workflow system for Student:

Once the student submits their live advisement form, they are presented with a workflow page. That shows them the process of their live advisement form, and notifies them when their advisor/advisors have advised them and they are eligible for enrollment.

View fully Rendered Front end in Appendix C: AdviseMe Implemented Frontend → Student View.

D. Faculty Version:

The Faculty Version of AdviseMe, has four major features that we implemented:

1) List of students in Need of Advisement:

This is the Faculty Advisor Main Page, on the left is a small profile of the user, displaying their name, EMPLID, and role. On the right side they see the list of students in need of advisement by semester on a first come, first serve order,

staying true to how advisement is done at CCNY. There is also a search bar where they can use the student's EMPLID to find the student's live advisement form. Once the advisor click on view advisement form for the student. They are now able to see basic information about the student, their live advisement form and the courses they intend to take the following semester, as well as the student's transcript. The advisor is able to advise the student by answering the advising questions that are directly from the 1st page of the CCNY advisement page. Once the student is approved, on the student version the workflow is updated. And the notes are displayed on the student view as well. Our application is fully dynamic and displays updated information in real time.

2) Archive:

We created a page specifically for faculty to have an archive of all the students from past to present advising notes. It serves to document all the student's information each semester.

3) Edit Advisement Form:

Faculty is able to make changes and edit the live advisement form/computer science curriculum. Any changes made are reflected throughout the entire application, on the checklist, live advisement form, and even course inform form (where we collect students grades). Overall, this feature allows faculty to makes updates, and for any changes in the curriculum to be updated and reflected easily on AdviseMe.

4) Edit Workflow:

Faculty is able to edit and update the permissions needed for students above or below 45 credits. For students who are below 45 credits they are required to be advised by an Academic Advisor, while those above 45 credits are required to be advised by both Faculty and Academic Advisors. With this feature, advisors can update/edit the workflow and it will be reflected on the entire platform. Also faculty can view previews of the workflow to see what the student will view.

View fully Rendered Front end in Appendix C: AdviseMe Implemented Frontend → Faculty View.

E. Special Features:

1) View Transcript and Curriculum:

Students and Faculty can view Transcripts and Curriculum on our website, and download the files. It's a great way to display documents, without having to leave AdviseMe.

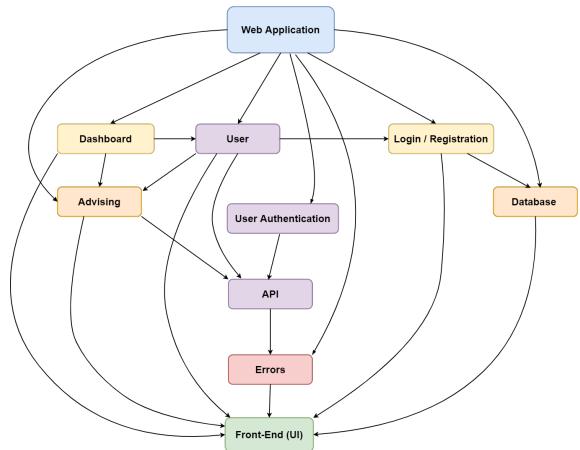
2) Download List of Graduates:

A special feature we added specifically for the faculty version is a list of upcoming graduates. On this page you can download individual students transcript/info. And also download the entire list of upcoming students as a excel or csv file. On these files you can see the list of student names, EMPLIDs, email, and more basic information. It's a great way for faculty to reach out to incoming graduates with important information and deadlines.

View fully Rendered Front end in Appendix C: AdviseMe Implemented Frontend → Special Features.

VII. PROJECT ORGANIZATION

A. Module Dependency



The web application depends on User's existence. Users depend on registration to populate the database. The web application also depends on the database to load content previously committed and stored. Web application depends on the Dashboard. The Dashboard will be different depending on the user's role, so the Dashboard depends on the User module. The User and Dashboard modules depend on the advising module as the Dashboard needs to navigate it, and the user needs to get advising done. The Advising and User module depends on the API module to make all get and post requests. The web application depends on the authentication module to enforce security rules. This module also depends on the API module as individual get and post requests may require specific user roles. The web application and all API's can throw run time or permission errors and depend on the Errors module to gracefully handle them. All of the modules depend on the Front-End (UI) module to correctly display all the dynamic site behavior when data is flowing, and when get and post requests are made.

B. Project Timeline

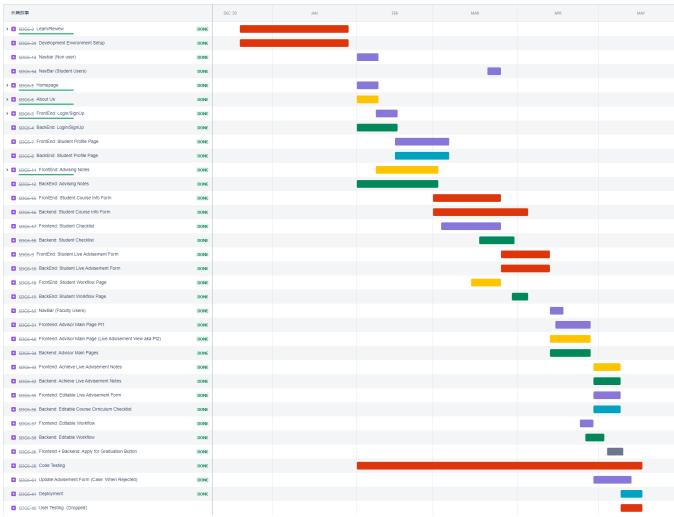
Agile software development refers to software development methodologies based on iterative development, where requirements and solutions evolve through collaboration. Scrum is a subset of Agile. A Scrum process is distinguished from other agile processes by specific concepts and practices, divided into the three categories of Roles, Artifacts, and Time Boxes.

Jira Software is an agile project management tool that supports any agile methodology, such as Scrum, Kanban. Jira provides agile boards, Roadmap (events and timeline), code (version control interface that connects to Github), and Pages (Project docs).

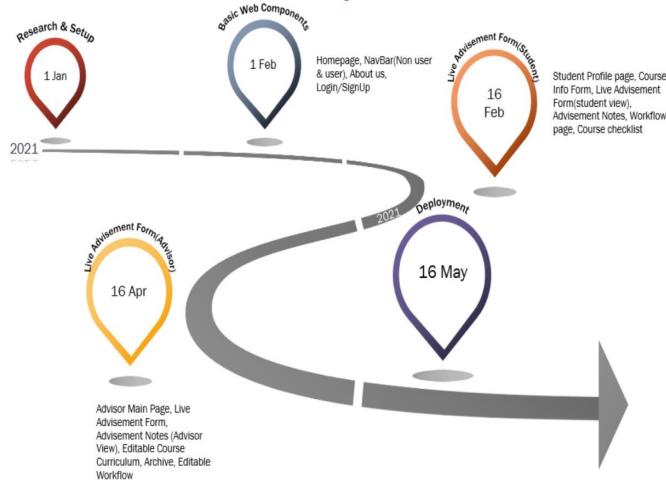
We break the project into weekly sprints based on features to accommodate school hours since all of us are full-time students for the semester. During winter break, all members research and getting acquainted with the TechStacks used for the project. Our primary goal for the winter break is to set up the development environment to get into implementation as the semester begins.

Weekly sprints will officially begin on February 1st and end on May 17th. We will spend our first two sprints implementing the website's basic components, such as the homepage and the Login/SignUp page. Then, we can focus on our primary feature, the Live Advisement form. We finished the student views of the Live Advisement form by April 7th, before the midterm. For the rest of the time, we implemented all the pages in the Advisor view, which includes the advisor main page, Advisor view of live advisement form, Editable course curriculum, and Editable workflow. Toward the end, to achieve a complete user experience, we added a Revise advisement form for students to change selections when they

got rejected, and an Advisement Archive for advisors to check their advisor notes in the past. On May 17th, we have finished deployment. Due to time constraints, we had to drop user testing.



C. Milestone



We started in January, research and getting acquainted with the TechStacks used for the project. Our project officially started in February, for the first two sprints, we finished implement the website's basic components, such as the homepage and the Login/SignUp page. Then, the student views of the Live Advisement form for the next 6 sprints. For the rest of the time, we implemented all the pages in the Advisor view.

D. Evaluation

To evaluation this website, we planned to invite ten students, two academic advisors, and two faculty advisors to try out the website. They will be given tester accounts across various roles and evaluate the website based on: UI completeness; can a user navigate over the webpage without external assist; Can each user role finish their assigned works within the website at ease; Advantage and disadvantage of the website in comparison to DegreeWorks; Possibility for this website to replace DegreeWorks for good.

VIII. DISCUSSION - LIMITATIONS AND CHALLENGES

We have experienced a lot of challenges and limitations throughout the development of the application. Our first challenge appeared while designing the course info form. The course info form is created to gather the completed courses from students when they first sign in.

However, the Course Info form requires students to go to a separate page each time a new grade needs to be added for any course. A regular Select Field (not Query Select Field) for grade choices, wrapped into a FieldList (which is a library in flask-wtf) would allow us to show all the courses with a nice drop down to insert a grade. This difficulty makes it challenging to create a user-friendly user interface that salvages students' time. With more time, we would remodify our course info form by using QuerySelectMultipleField (which is also a library in flask-wtf), it would perfectly address this issue.

Our website application is also limited to an MVP prototype and is not ready to deploy. This was planned in the earlier planning phase. However, deployment will be tough due to various challenges and an ambitious schedule. One of the major challenges for deployment is site security of confidential student information. Encryption is the best way to prevent unauthorized access to your data, and it may be described as the transformation of the data into an alternate format that a person who has access to a decryption key may read exclusively. Although we encrypt some information from students such as passwords, we do not have enough time to finalize all site security of confidential student information. File Storage limitations with most cloud hosting providers are another major challenge that we face to deploy our web. There are various types of files that need to be saved for each student in the database, such as student images and transcript pdf files. So, we do require sizable file storage for saving data in the cloud. Deployment of our web services offers another challenge to our team. We recognized SQLite backend not compliant with PostgreSQL in some areas. But still, we could figure out a better solution by doing more research.

IX. CONCLUSION

The purpose of our project is to create a user-friendly and well-organized advising platform to assist CCNY students in developing the academic plans and automating the live advisement. After completing the major features of our platform, we hope that some useful functionalities can be created with the given time, such as GPA calculator and advising deadline email reminder. Furthermore, we expect CCNY with the facilities will cautiously attempt to carry out our platform to help CCNY students automate the advisement so that we may refine it further with user's feedback. With the future research, more features we expect to add into our platform to better improve the user's experience, for instance, we can generate an AI chat bot to help navigate the site and answer the FAQ. As David Heinemeier Hansson stated that, "The best way to predict the future is to implement it." Doing advisement efficiently is the way that students can success in college. We hope our web application can actually help our CCNY students make better academic decisions in the future.

X. AUTHOR CONTRIBUTIONS



Chrystal Mingo, is a senior majoring in Computer Science at The City College of New York. She has interned at various companies such as Verizon, Morgan Stanley, and Citi as a software developer. She also has a passion for teaching and has worked as a Teacher for the Girls Who Code Summer Immersion Program, and taught Algebra I and II, as well as Intro and AP

Computer Science courses in Spanish and English at Gregorio Luperon High School. Chrystal Mingo is also a Grace Hopper 2019 Scholarship Recipient, and current President of Women in Computer. She will be graduating from CCNY in Spring 2021 and entering Citi's EIOT full-time rotational program as a project manager. She is one of the frontend developers for AdviseMe and believes this project will be a game-changer for advisement at CCNY.



Rehman Arshad, is a senior majoring in Computer Science at The City College of New York. He has interned at various academic research institutions at The Groove school of Engineering, such as NOAA Crest (National Oceanic Atmospheric Administration) and Professor Tarek Sadawi. At NOAA Crest, he worked on data collection and analysis using data from the National Weather Service. With Professor Tarek Sadawi he worked on an IoT medical application that would enable remote patient monitoring as a research assistant. He was a part of CUNY Tech Prep cohort 5, a Full Stack program part of NYC Tech Talent Pipeline. Also, he has a fascination with mathematics and a passion for computer graphics and video game development. He will be graduating from The City College of New York in Fall 2021. He is one of the backend developers for AdviseMe and hopes this online platform can help the next generation of students entering CCNY.



Xunshan Lin, is a senior majoring in computer engineering at The City College of New York. Previously he was interning as a Python programmer at Mini circuits company. He is beginning to pick up an interest in full-stack development at his senior design project at CCNY. He works as one of the backend developers for AdviseMe and hopes this platform will help students make better academic plans at CCNY.



Zhicong Wen is a senior majoring in Computer Science at City College of New York. Previously he was interning as a Python programmer at the NYC Department of Environmental Protection(DEP). He also worked on Quality Assurance of the New York City sewer system database. He will be graduating from CCNY in Fall 2021. He is one of the frontend developers for AdviseMe.

XI. ACKNOWLEDGMENTS

The authors would like to thank our Professor, Huy Vo, for mentoring our project and helping us solidify our game plan for Spring 2021. Also, thank you to our peers for all of their feedback towards the failures of advisement at CCNY. Thanks to your feedback we were able to build a strong design plan, and select the core functionalities to make our online advisement platform, the best for our users!

REFERENCES

- [1] Atlassian. “Get Started with Agile Project Management.” Atlassian, www.atlassian.com/agile/project-management.
- [2] AWS Documentation. docs.aws.amazon.com/index.html.
- [3] “CUNY City College Graduation amp; Retention.” College Factual, 13 Nov. 2020, www.collegefactual.com/colleges/cuny-city-college/academic-life/graduation-and-retention/.
- [4] www.cuny.edu/about/administration/offices/registrar/resources/degreeworks/.
- [5] “Figma to HTML.” Figma, www.figma.com/community/plugin/851183094275736358/Figma-to-HTML.
- [6] “Flask.” Pallets, palletsprojects.com/p/flask/.
- [7] “What Is Python? Executive Summary.” Python.org, www.python.org/doc/essays/blurb/.

APPENDIX A APPLICATION PROTOTYPE

A. Student View of AdviseMe

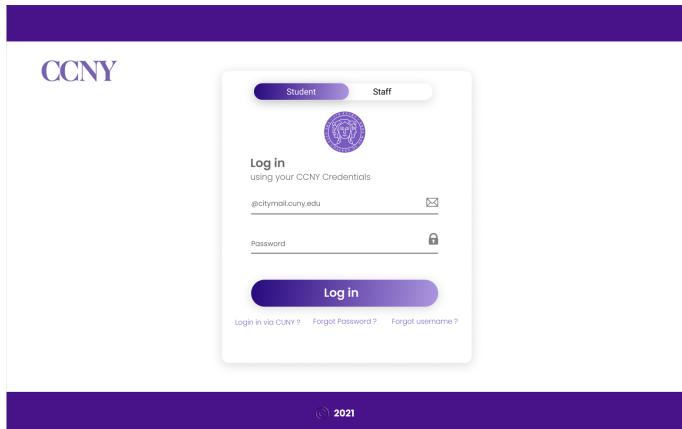


Figure 1: Student Login

All students will have the ability to log in. As shown above students will log in using their CCNY credentials.

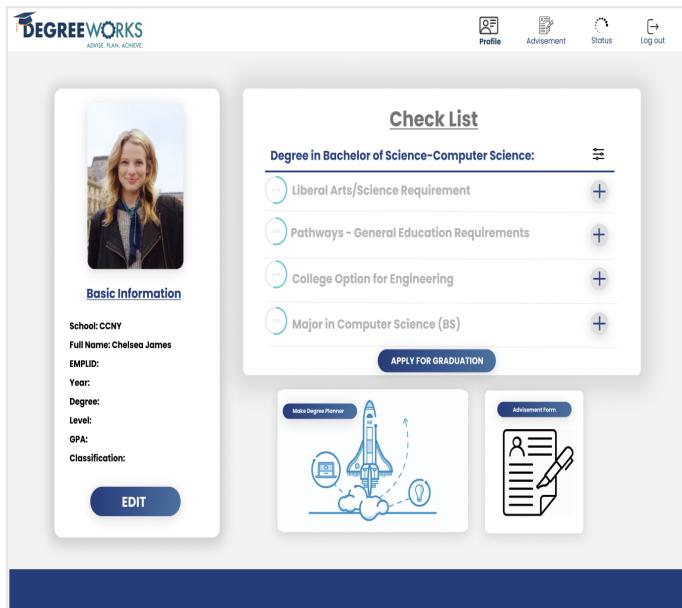


Figure 2: Student Profile

Upon logging into the platform, students will encounter the home page, which is called Profile. Students will be able to see their profile with basic information such as their EMPLID, year, school, GPA, and edit to add more content such as bio. Students can also see their checklist, which showcases their progress within each requirement needed to graduate. Only students who have reached all the requirements can then press to apply for graduation. There are also direct links to a personalized degree planner and our live advisement form.

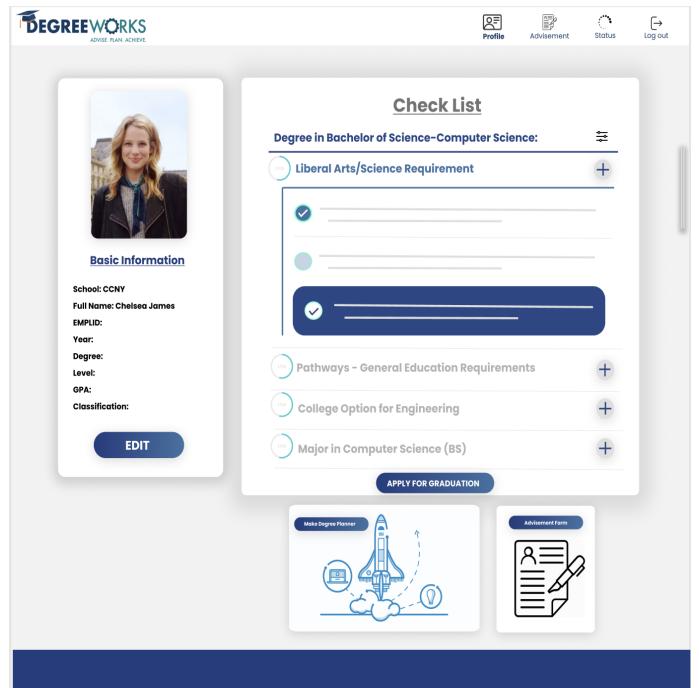


Figure 3: Student Profile: When students view courses under checklist requirements

This is the view from the Advisement tab. Students will be able to select advising notes or the advisement form (using a drop-down menu).

The live advisement form has automatically filled all the courses the student has taken grades needed that apply to the major/minor. And checks off the classes the student is currently taking. Allows Students to check mark courses they intend on taking the following semester. Will enable students to submit an advisement form directly to the advisor upon clicking submit. This makes filling out advisement at CCNY quick and easy.

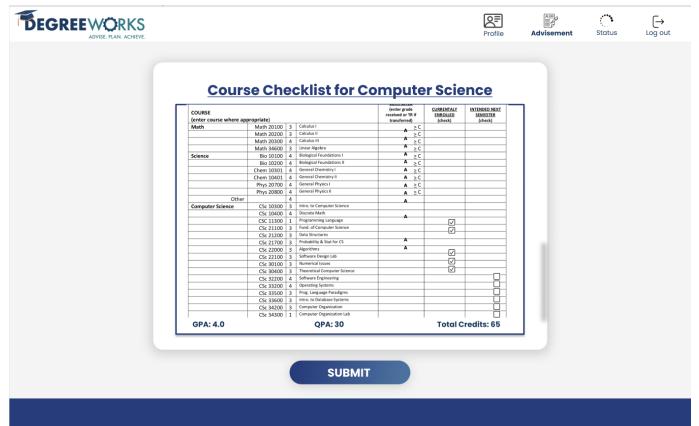


Figure 4: Student Live Advisement and as you scroll down view of GPA, QPA, and Credits

Students can also view their advising notes from previous semesters and get a notification when their advising notes for the semester have been submitted. We formatted the advising notes so that students can select what semester they'll like to view. Also, in the Computer Science department, below 45 credits only requires advising by the academic advisor, and students with over 45 credits get advised by both a faculty and academic advisor.

Figure 5: Advising Notes

The students can also view their workflow. This Status tab displays approval/steps of automated advising and notifies when a student is eligible for enrollment. Students have two viewing options to see the progress of their process.

Figure 6: Status

B. Advisor/Staff View

The second version of the platform we will delve into is the advisor/staff version:

All staff will have the ability to log in. As shown below, advisors will login in using their CCNY credentials.

Figure 7: Staff Login

The main page for the advisors/staff is the profile and the list of students in need of advisement.

Figure 8: Staff Main Page

When an advisor clicks on the plus sign, they can see the student's info and click to view the student's advisement form. In this form, the adviser can see the student's live advisement form and write their advising notes. They can also approve or decline the student's form and then submit their decision.

Figure 9: Student Advisement Form View from Advisor Page

The staff can also edit the Computer Science Curriculum (live advisement form).

The staff can also edit and save the workflow. They can select what permissions are needed depending on credits and grades change.

Figure 10: Edit Advisement Form Page

The staff can also edit and save the workflow. They can select what permissions are needed depending on credits and grades change.

The staff can also edit and save the workflow. They can select what permissions are needed depending on credits and grades change.

Figure 11: Staff Workflow View Page

APPENDIX B DEVELOPMENT PLANS

A. Database Schema

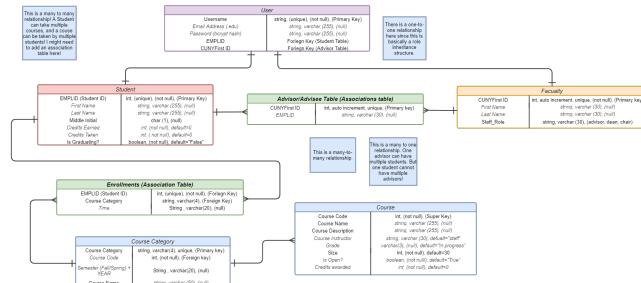


Figure. 1 - A comprehensive breakdown of our relational database is as follows: The User table is inherited by two other tables: the Student table and the Faculty Table. The Faculty table can consist of Academic Advisors, Faculty Advisors, the Dean, and the Department Chair. There is a many-to-many relationship between faculties and students because one advisor can advise multiple students. A student can have more than one advisor throughout their academic career. If a student has less than 45 credits completed, they rely on an academic advisor. Otherwise, they must rely on Faculty advisors. Student also has a many-to-many relationship with the Course Category table. To simplify matters, we can view the Course Category and the Course table as a single entity. So students can enroll in multiple courses, and numerous students can take one course. The Course Category table has a one-to-many relationship because a course category can have multiple courses, but not the other way around.

B. User Authentication

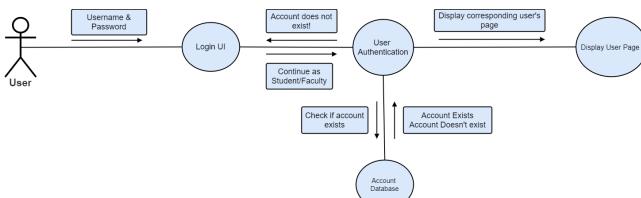


Figure. 2 - A user provides their username and password when logging in. The system checks whether or not a user with such an account exists in the database. If the user does not exist, it will throw an error. If the user exists, the system will check if the password matches the crypt hash stored in the database. If the hash values match, the user is successfully authenticated and logged in. Once logged in, the user (depending on their user role) will be re-directed to their respective profile page.

C. Student Management

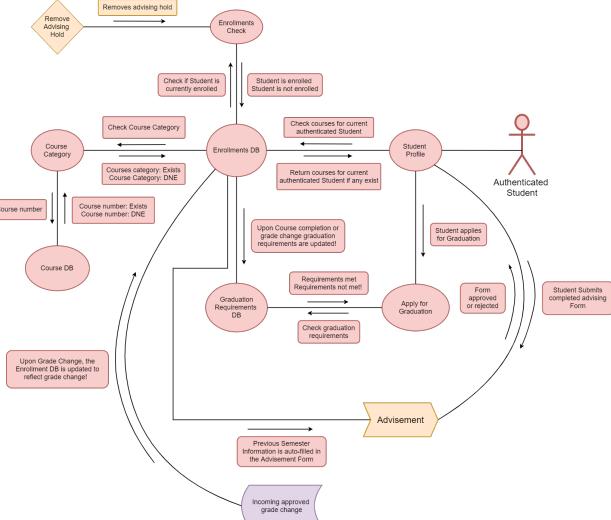


Figure. 3 - A Student is redirected to the student profile. The student can check courses from the enrollments database. The enrollments database is linked to the course category database which is connected to the course database. Upon course completion, the graduation requirements database is updated. If the student meets all requirements, the student can apply for graduation. Otherwise, the button will not display. A Student can also submit a completed advisement form, which an advisor can approve or reject. The enrollment database pre-fills the advisement form with information from previously completed semesters. Grade changes approved by the Dean and Chair updates the enrollment database and any other related database such as the graduation requirements database, which will check for updates and see if requirements are met. When an advisement form is approved, the advisement check is updated to indicate the student has finished advising.

D. Advisor Management

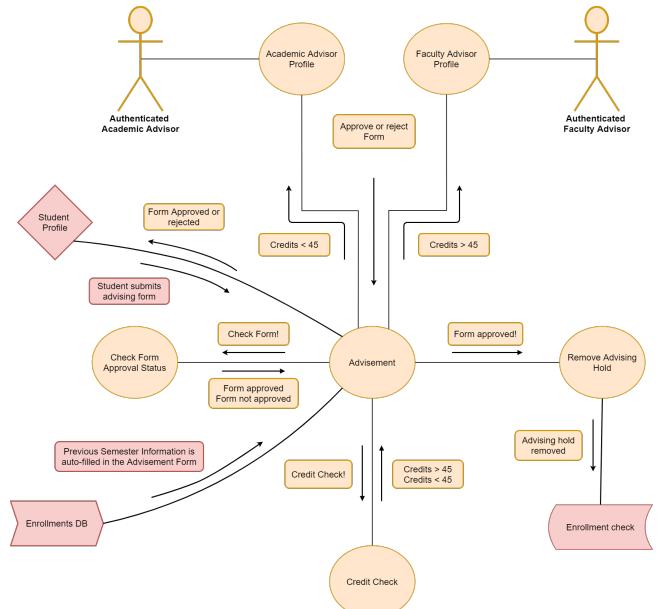


Figure. 4 - An advisor is redirected to each advisor's respected profile. The advisor can see all completed advisement forms submitted by students. The advisement form is pre-filled with previous semester information. They can approve or reject a Students advising form, fill out advisement notes, and when approved the Student's advisement hold is removed. If the Student is under 45 credits they are directed to an academic advisor. If the

Student is over 45 credits they are directed to a faculty advisor. If an advisor rejects an advisement form, (due to not completing pre-requisites) then the Student must re-submit a completed advising form.

E. Dean and Chair Management

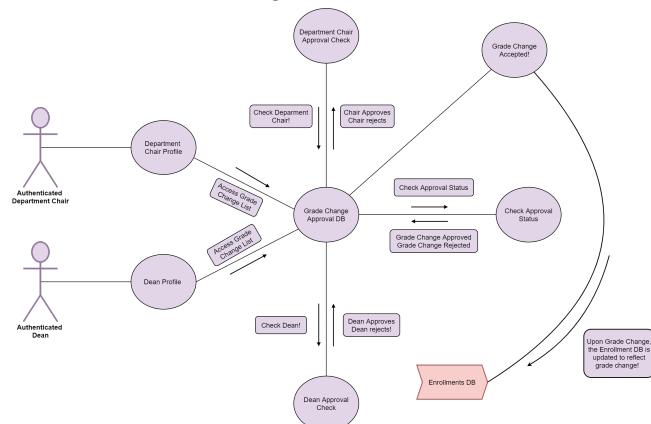


Figure. 5 - The Dean and department chair can view the list of all students whose grades are changed. They can approve or reject the grade change. The grade change status will check whether or not the grade is approved. For a grade change to be approved, both the Dean and the Chair need to approve a grade change. Only then will the grade change be approved. Once approved, the Enrollment database is updated with the newest grade for the specific course within a category.

F. High Level Architectural Diagram

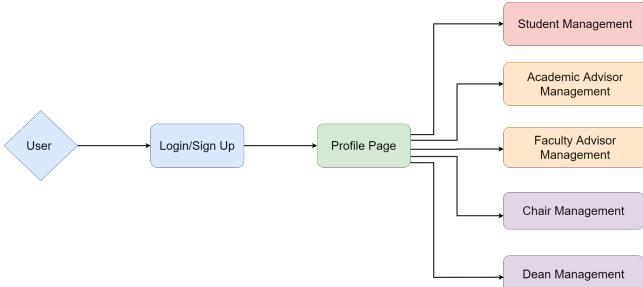


Figure. 6 - A more general architecture diagram showcasing the different components that need to be implemented.

G. Core Architectural Diagram

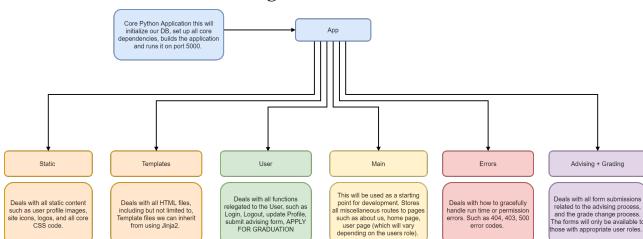


Figure. 7 - This is a more detailed core architecture diagram. It shows us the six core modules of this system. By default, the flask framework has a Static and Template module. In the static module, all static files such as images, profile photos, the site logo, and core CSS would fall into this module. The templates module would contain all the core HTML files for our application. The User module would hold all functions relegated to the user, such as login/logout, update profile, submit advising form, and apply for graduation. The main module consists of all miscellaneous routes, such as the about us page. The Error module consists of graceful

error handling for permission or server errors such as 404, 403, and 500. The Advising and grading module deals with all functions relegated to the advising functions.

H. Complete Use Case Diagram (Small)

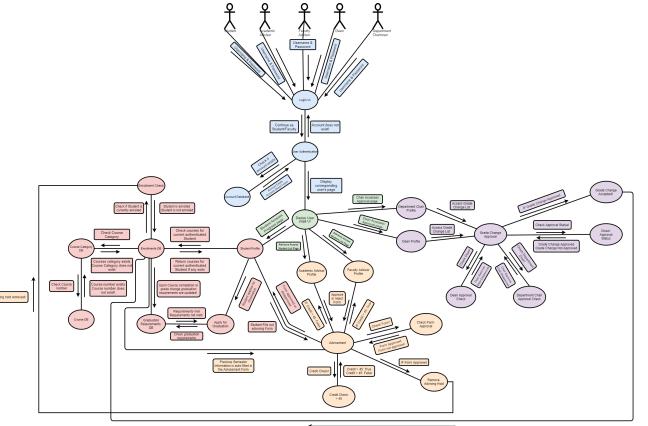
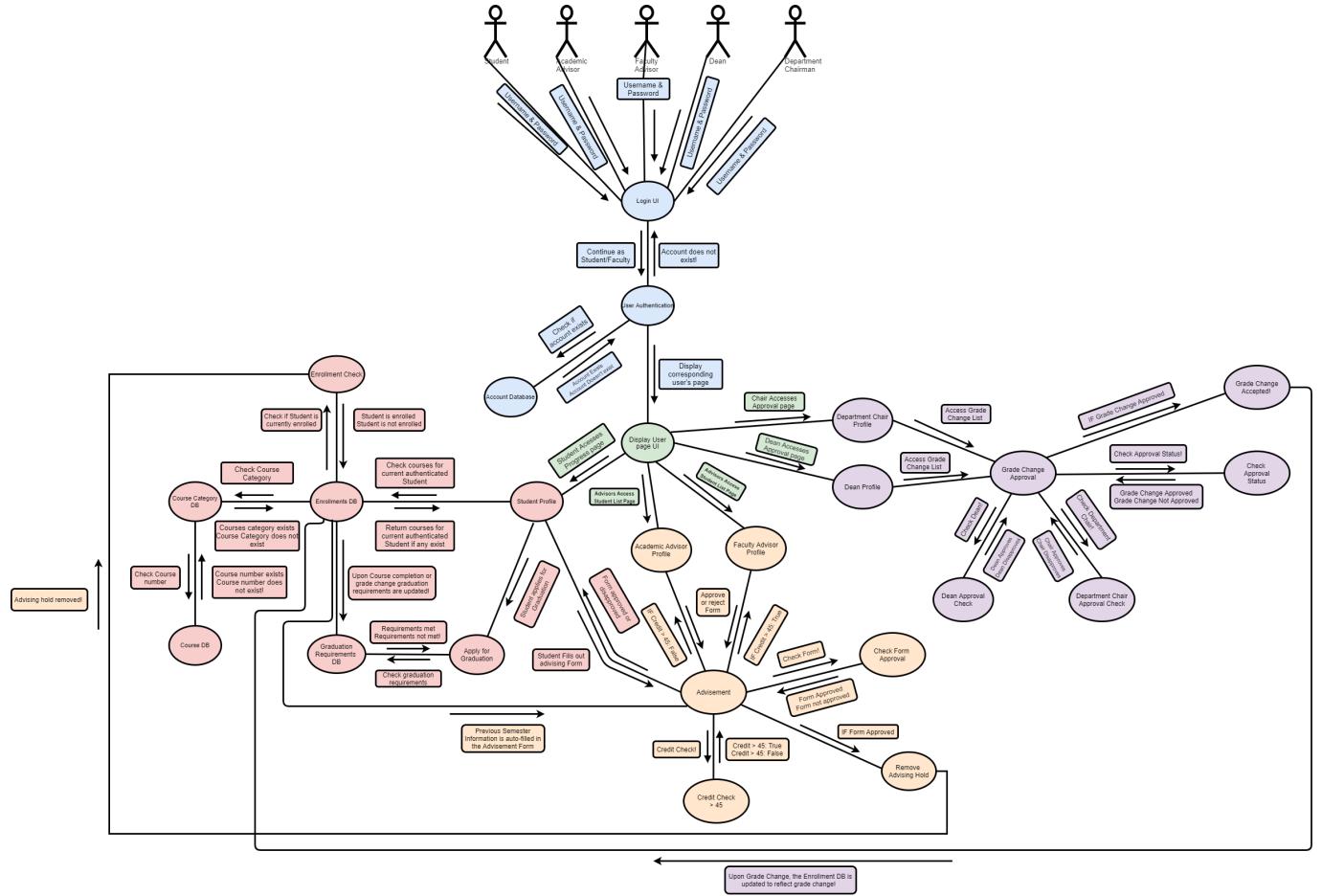


Figure. 8 - This diagram is a combination of (figures 2 - 5)

I. Complete Use Case Diagram (Large)



An enlarged version of **Figure. 8** in Appendix B.

APPENDIX C ADVISEME IMPLEMENTED FRONTEND

A. Homepage Frontend

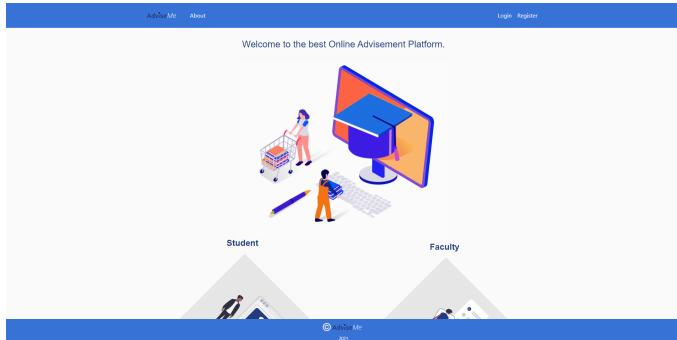


Figure. 1 - Homepage

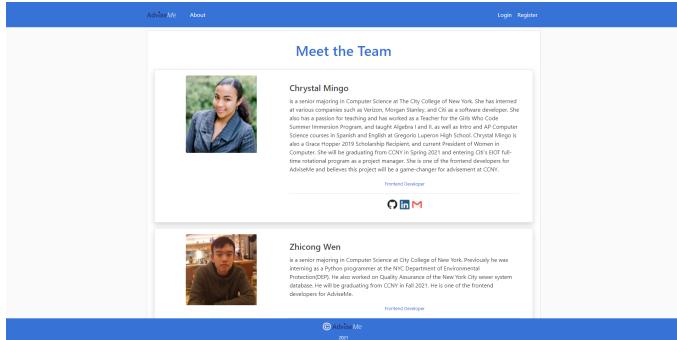


Figure. 2 - About Us

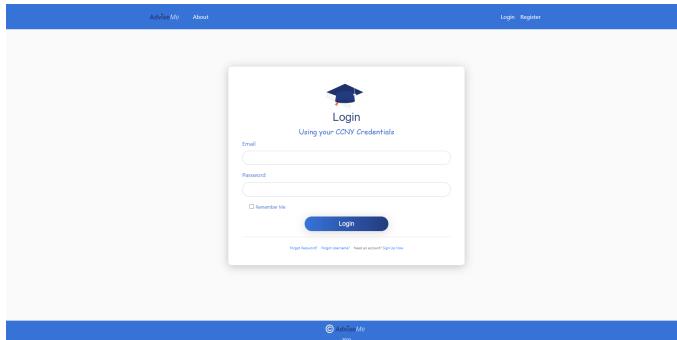


Figure. 3 - Login

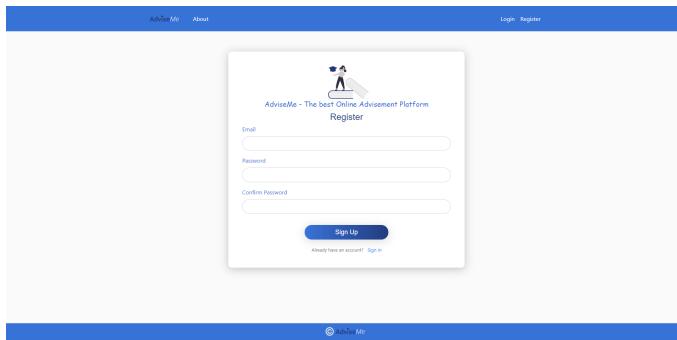


Figure. 4 - Register

B. Student View

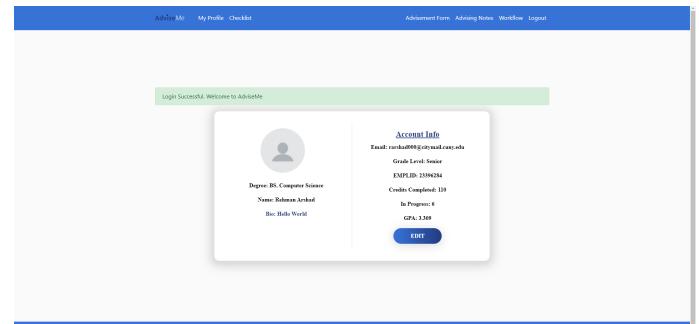


Figure. 5 - Student Profile

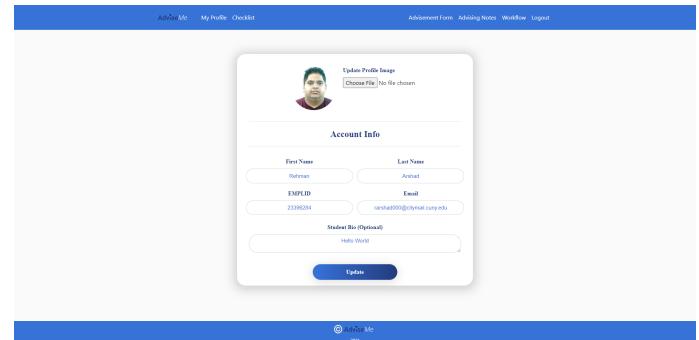


Figure. 6 - Student Profile Edit

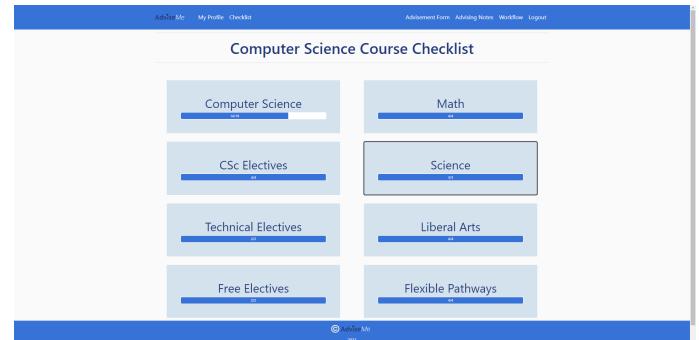


Figure. 7 - Checklist

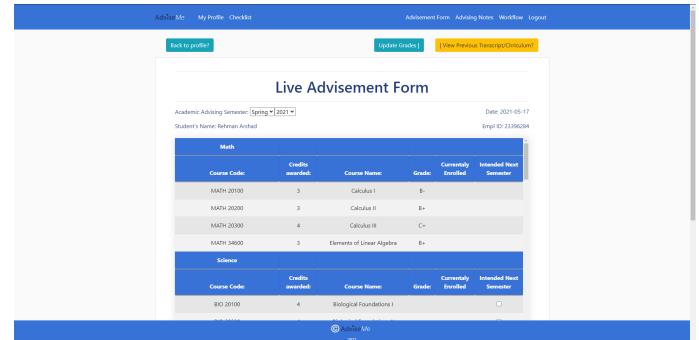


Figure. 8 - Live Advisement Form

The screenshots show three different sections of the Advising Note template:

- Notes from Your Advisor:** A decorative header with a graduation cap icon. Below it is a blue header bar with "Advising Note" and "2021-05-17". The main content area contains a large blue box labeled "SPRING 2021" and a smaller box below it.
- Academic Advising Note:** A section titled "Academic Advising Note" with a blue header bar. It contains a list of statements and checkboxes:
 - 1. I have reviewed the student's transcript and discussed academic performance. **Comment:** Student is performing adequately.
 - 2. I have reviewed the student's intended course schedule for next semester. **Recommended changes/Comments:** The course progression looks good!
 - 3. The student has been advised that the course schedule for the upcoming semester is based on his/her satisfactory completion of course: **No.**
 - 4. I have reviewed the student's intended course schedule for next semester. **Tutorial Services:**
Counseling/Psychological, Financial Personnel etc:
 - Faculty Certificate/Office Head:**
- Faculty Advising Note:** A section titled "Faculty Advising Note" with a blue header bar. It contains a list of statements and checkboxes:
 - Recommended changes/Comments:** This course progression looks good!
 - 3. The student has been advised that the course schedule for the upcoming semester is based on his/her satisfactory completion of course: **No.**
 - 4. I have reviewed the student's intended course schedule for next semester. **Tutorial Services:**
Counseling/Psychological, Financial Personnel etc:
Faculty Certificate/Office Head:
Career Advisee:
Scholarships:
Internship Opportunities:
Follow-up Advisement Sessions:

Figure. 9-10 - Advising Notes View

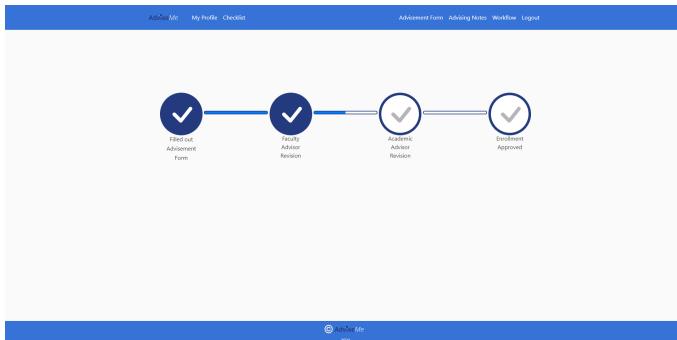


Figure. 11 - Workflow

C. Faculty View

The screenshot shows the "Students in Need of Advisement" section of the Faculty Main Page. It includes a search bar and a list of students:

- School: CUNY - CCNY
- Role: Academic Advisor
- Name: Rehman Arshad
- EmpID: 13346285
- QPA: 2.4

Figure. 12 - Faculty Main Page

The screenshots show four different sections of the Advising Note template:

- Course Checklist for Computer Science:** A table showing course details:

Course Code	Credit hours	Course Name	Grade	Currently Enrolled	Intended Next Semester
MATH 20100	3	Calculus I	B-		
MATH 20200	3	Calculus II	B+		
MATH 20300	4	Calculus III	C+		
MATH 34000	3	Elements of Linear Algebra	B+		
- Faculty Advising Note:** A section titled "Faculty Advising Note" with a blue header bar. It contains a list of statements and checkboxes:
 - 1. I have reviewed the student's transcript and discussed academic performance. **Comment:** Student is performing adequately.
 - 2. I have reviewed the student's intended course schedule for next semester. **Recommended changes/Comments:** The course progression looks good!
 - 3. The student has been advised that the course schedule for the upcoming semester is based on his/her satisfactory completion of course: **No.**
 - 4. I have reviewed the student's intended course schedule for next semester. **Tutorial Services:**
Counseling/Psychological, Financial Personnel etc:
- Academic Advising Note:** A section titled "Academic Advising Note" with a blue header bar. It contains a list of statements and checkboxes:
 - Comments:** This looks good! Just make sure to do well in paradigm!
 - Additional:** Make sure to pass paradigm as this is your second time taking it, since you dropped it the first time. Any third attempt will require departmental consent and place you on academic probation.

Figure. 13-14 - Academic Advisor - Advising Notes View

The screenshot shows the "Your Student's Advisement Record" section of the Archive View. It displays a summary of the student's advisement record:

- Student Name: Arshad, Rehman
- Semesters: SPRING 2021, FALL 2021
- Dates: 2021-05-17, 2021-05-17

Figure. 15 - Archive View

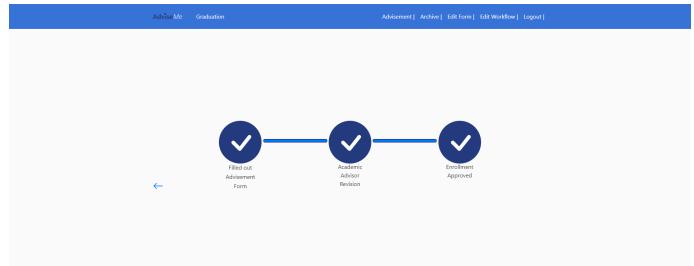
[Add New](#) [Graduation](#)

[Admission](#) | [Archive](#) | [Edit Form](#) | [Edit Workflow](#) | [Logout](#)

Edit Computer Science Curriculum

Add new Course to Curriculum?

Course Serial	Course Name	Course Department	Credits	Designation	Course Description
CSC 12000	Intro to CS program	CSC	3	Core Requirements	Learn C++
CSC 10400	Discrete Math	CSC	4	Core Requirements	Fundamental mathematics for CS
CSC 11300	Intro to Python	CSC	1	Core Requirements	Python
CSC 21100	Fundamentals to CS	CSC	3	Core Requirements	Understanding binary logic
CSC 21200	Data Structures	CSC	3	Core Requirements	Data Structures for algorithms
CSC 21700	Probability and Stats to	CSC	3	Core Requirements	Learn Statistical Analysis



[Add New](#) [Graduation](#)

[Admission](#) | [Archive](#) | [Edit Form](#) | [Edit Workflow](#) | [Logout](#)

[Edit an existing Course in the Curriculum?](#)

[Add New](#) [Graduation](#)

[Admission](#) | [Archive](#) | [Edit Form](#) | [Edit Workflow](#) | [Logout](#)

Course ID:	Course Serial:	Course Name:	Course Department:	Course Description:	Credits:	Designation:	Entered:
#	CSC	Intro to CS	CSC	Learn C++	Core Requirement	3	Delete
1	CSC 12300	Intro to CS programming	CSC	Learn C++	Core Requirement	3	Delete
2	CSC 10400	Discrete Math	CSC	Fundamental mathematics for CS	Core Requirement	4	Delete
3	CSC 11300	Intro to Python	CSC	Python	Core Requirement	1	Delete
4	CSC 21100	Fundamentals to CS	CSC	Understanding binary logic	Core Requirement	3	Delete
5	CSC 21200	Data Structures	CSC	Data Structures for algorithms	Core Requirement	3	Delete
6	CSC 21700	Probability and Stats to CS	CSC	Learn Statistical Analysis	Core Requirement	3	Delete
7	CSC 22000	Algorithms	CSC	Understanding the use of Data Structures for efficiency	Core Requirement	3	Delete
8	CSC 22100	Software Design Lab	CSC	Use Java OOP to make GUI	Core Requirement	3	Delete

[Add New](#) [Graduation](#)

[Admission](#) | [Archive](#) | [Edit Form](#) | [Edit Workflow](#) | [Logout](#)

#	Course ID:	Course Name:	Course Department:	Course Description:	Credits:	Designation:	Entered:
93	ANTH 20100	Cross Cultural Perspectives	ANTH	Learn about different social norms and perspectives	350(2000)	3	Delete
94	EDCE 20000	Language, Mind, and Society	EDCE	Varies by Instructor	350(2000)	3	Delete
95	ANST 20100	The Holocaust	ANST	Experience survivor stories of the past, and reflect on why you would do if you don't need to answer. All questions are theoretical!	350(2000)	3	Delete
96	HIST 24000	The United States From Its Origins to 1877	HIST	Colonial Era to 1877	350(2000)	3	Delete
97	HIST 24100	The United States since 1865	HIST	The aftermath of the Civil war reconstruction, Jim Crow, WWI, Civil Rights Movement, Cold War, to present	350(2000)	3	Delete
98	EAS 21700	Systems Analysis of the Earth	EAS	Analysis and modeling of plate tectonics and climate change	Technical Elective	4	Delete
99	EAS 22000	Structural Geology	EAS	Geometry of elementary earth structures	Technical Elective	4	Delete
100	EAS 30000	ESS Modeling: Databases	EAS	Modeling of global and local environmental problems	Technical Elective	3	Delete
101	PHY 20300	General Physics I	PHY	Introduction to Physics for science majors and pre-med/bio majors	Technical Elective	4	Delete
102	PHY 20400	General Physics II	PHY	Intermediate Physics for the science majors and pre-med/bio majors	Technical Elective	4	Delete

Figure. 16-18 Curriculum Edit Form



Figure. 20 - Preview Workflow below 45



Figure. 21 - Preview Workflow above 45



D. Special Features

[Add New](#) [Graduation](#)

[Admission](#) | [Archive](#) | [Edit Form](#) | [Edit Workflow](#) | [Logout](#)

Edit Workflow

For students under 45 credits:

- Taken Admisse Form
- Faculty Advisor Approval
- Academic Advisor Approval
- Eligible for Scholarship

For students above 45 credits:

- Taken Admisse Form
- Faculty Advisor Approval
- Academic Advisor Approval
- Eligible for Scholarship

[Update](#)

[Add New](#) [Graduation](#)

[Admission](#) | [Archive](#) | [Edit Form](#) | [Edit Workflow](#) | [Logout](#)

City College Hall of Alumni

FALL 2021

#	Student	EMPID	GPA	GPA	Transcript
1	Clark Kent	23348377	3.3	12	Download Transcript
2	Steve Jobs	23353464	3.1	10	Download Transcript
3	Bill Gates	23356744	4	33	Download Transcript
4	Kareem Ibrahim	23367137	3.6	28	Download Transcript
5	Gabe Newell	23368089	4	33	Download Transcript
6	Steve Wozniak	23368499	3.4	14	Download Transcript
7	Todd Howard	23376007	3.532	30	Download Transcript
8	James Bond	23380007	3.42	12	Download Transcript
9	Bruce Wayne	23394232	3.32	13	Download Transcript
10	Ivan Connor	23396488	3.47	26	Download Transcript

[Export Data as CSV File](#)

[Export Data as Excel Sheet](#)

Figure. 19 - Edit Workflow

Figure. 22-27 - Graduation List

A1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
0	23344077	Clark	Kent	3.3	12														
1	23354844	Steve	Jobs	3.1	10														
2	23354844	Bill	Gates	3.1	10														
3	23356737	Kareem	Ibrahim	3.6	28														
4	23356932	Gabe	Newwell	4	33														
5	23356932	Bruce	Wayne	3.32	13														
6	23356999	John	Connor	3.42	28														
7	23356999	James	Bond	3.42	12														
8	23356999	Albert	Einstein	3.23	16														
9	23360007	Albert	Turing	4	35														
10																			
11																			
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			
21																			
22																			
23																			
24																			
25																			
26																			
27																			
28																			
29																			
30																			
31																			
32																			
33																			
34																			

Figure. 28-29 - Transcript Overview

The latest version of the curriculum sheet aggregates any contracts and preprobable information in the undergraduate catalog.