

Epic Id	Epic
1	User Authentication
Description	Implement user authentication

user story id	user story name
1	User registration
Description	As a Site User I can register on the website and create an account so that I can log into the website and use it.
Story points	8
MoSCoW Category	Must Have
Epic.User story Dependencies	N/A
Task id	Tasks
1	Install Django, gunicorn, dj_database_url and psycopg2
2	Create Music Aid project and Songbook app
3	Set up clouinary and elephant SQL
4	Deploy to Heroku
5	Code header and footer in base.html view and templates
6	Code landing page view and templates
7	Code navigation view and templates
8	Install allauth
9	Configure allauth and Django for user / Admin registration as necessary
10	Style the new front end features
11	Test the new functionality
Acceptance criteria id	Acceptance criteria
1	The user can register an account on the website
2	<b>The following acceptance criteria for feature id 1 'Landing Page' have been satisfied</b>
2.1	The landing page has a navigation menu
2.2	The landing page has a header that is fixed to the top of the page.
2.3	The landing page has a footer that is fixed to the top of the page.
2.4	The landing page has a navigation menu
2.5	The header and the footer have the company name and application name displayed.
2.6	Clicking the company name or application name will take the user to the Landing page
2.7	If logged in, the landing page displays the users songbook
2.8	The landing page is styled correctly for the site.
2.9	The landing page is responsive and resizes as necessary on a given screen.
3	<b>The following acceptance criteria for feature id '20' Navigation have been satisfied.</b>

3.1	The navigation menu has a navigation link to the user 'Registration' Page
3.2	The navigation menu has navigation link to the user 'login' Page that is styled correctly for the site
3.3	On successful login, the navigation menu has a logout link in place of the login link.
3.4	On successful Admin login, the navigation menu has an admin page link
3.5	The navigation menu is styled correctly for the site.
3.6	The site is responsive and the navigation menu becomes a 'burger icon' on small screens
4	<b>The following acceptance criteria for feature id 3 'Register' have been satisfied</b>
4.1	The user registration page has a 'Username' field that is a required field.
4.2	The user registration page has an 'Email' field that is optional.
4.3	The user registration page 'Email' field is validated and an invalid email address cannot be submitted.
4.4	The user registration page has a 'Password' and password confirmation field that are validated and that must match for submission to take place.
4.5	The user registration page has a 'Sign Up' button.
4.6	On successful registration there is a message to the user confirming they are logged in as that user.
4.7	The user registration page is styled correctly for the site.
4.8	The user registration page is responsive and resizes as necessary on a given screen.

user story id	user story name
2	User Sign In
Description	As a Site User I can sign into the website so that I can use it.
Story points	1
MoSCoW Category	Must Have
Epic.User story Dependencies	1.1
Task id	Tasks
1	Configure allauth and Django for user / Admin sign in as necessary
2	Style the new front end features
3	Test the new functionality
Acceptance criteria id	Acceptance criteria
1	The site user / admin can sign into the website with their previously registered credentials.
2	<b>The following acceptance criteria for feature id 2 'Sign In' have been satisfied.</b>
2.1	The 'Login' page has a link to the user registration page (for use in the even a user has not yet registered)
2.2	The 'Login' page has a username field that is a required field required.
2.3	The 'Login' page has a password field that is required field.
2.4	On successful sign in the user is notified that they have successfully signed in.
2.5	The login page is styled correctly for the site.
2.6	The login page is responsive and resizes for the screen on which it is being viewed.

user story id	user story name
3	User Sign Out
Description	As a Site User /Admin I can sign out of the website so that others cannot access my account from the device I have been using.
Story points	1
MoSCoW Category	Must Have
Epic.User story Dependencies	1.2
Task id	Tasks
1	Configure allauth and Django for user / Admin sign out as necessary
2	Style the new front end features
3	Test the new functionality
Acceptance criteria id	Acceptance criteria
1	The user can sign out of the website
2	The following acceptance criteria for feature id 6 'Sign Out' have been satisfied
2.1	On successful sign out, the user is notified that they have successfully signed out.
2.2	On clicking sign out, the user is asked to confirm that they'd like to sign out.
2.3	On successful logout the user is returned to the landing page.

user story id	user story name
4	User Deletion
Description	As a Site Admin I can delete users should a need arise.
Story points	1
MoSCoW Category	Must Have
Epic.User story Dependencies	1.1
Task id	Tasks
1	Configure the site admin page for user management
2	Test the new functionality
Acceptance criteria id	Acceptance criteria
1	The site admin can delete a user.
2	<b>The following acceptance criteria for feature id '4' Admin Page have been satisfied.</b>
2.1	The Admin page has a link to user admin features.
2.2	A site Admin can delete a selected users credentials from the user admin section of the site admin page.
2.3	On clicking to delete a user or users the site admin is asked to confirm that they'd like to delete those users.
2.4	On successful deletion there is a message to the site admin that the deletion has taken place.
2.5	The site admin clicking that they do not wish to confirm deletion returns them to the previous page.

Epic Id	Epic
2	Songbook
Description	Implement a way to manage song lyrics

user story id	user story name
1	Add a songs lyrics
Description	As a Site User I can add a songs title and lyrics and autoscroll speed and optionally the songs artist to my songbook so that I can view them later.
Story points	5
MoSCoW Category	Must Have
Epic.User story Dependencies	2.2
Task id	Tasks
1	Install summernote
2	Update the songbook list page with add button
3	Code the add song page view and templates
4	Configure summernote to have no toolbar and be used to enter text into the lyrics field.
5	Style the new front end features
6	Test the new functionality
Acceptance criteria id	Acceptance criteria
1	The site user can add a songs title and lyrics and autoscroll speed and optionally the songs artist to their songbook so that they can view them later.
2	<b>The following acceptance criteria for feature id '8' Songbook list have been satisfied.</b>
2.1	The songbook list page has an 'add' button that navigates to the add song page.
2.2	The add button is styled consistently with the rest of the site.
3	<b>The following acceptance criteria for feature id '7' Add Song have been satisfied.</b>
3.1	The add song page has the heading 'Song' just below the header that is styled consistently with the rest of the site.
3.2	The add song page has a title field that is validated, is required and allows up to 150 characters.
3.3	The add song page has an artist field that is validated, is optional and allows up to 150 characters.
3.4	The add song page has a text box lyrics field that allows for characters to be entered.
3.5	The add song page has a save button that saves the user entered data
3.6	The add song page has an exit button that returns the user to the songbook page.

<b>3.7</b>	The add song page has the regular site header, footer and navigation bar with their regular associated links (as defined in the full landing page (id 1) and navigation (id 20) feature acceptance criteria lists)
<b>3.8</b>	The add song page is styled correctly for the site.
<b>3.9</b>	The add song page is responsive and resizes for the screen on which it is being viewed.

user story id	user story name
2	<b>View a list of songs</b>
Description	As a Site User I can view a list of all the songs I've entered so I can make a selection.
Story points	3
MoSCoW Category	Must Have
Epic.User story Dependencies	1.1
Task id	Tasks
1	Update the navigation menu
2	Add song table to database model with title, artist, lyrics, user link and slug fields.
3	Code the songbook list page view and templates
4	Style the new front end features
5	Test the songbook list page
Acceptance criteria id	Acceptance criteria
1	The site user can view a list of songs that they have entered.
2	<b>The following acceptance criteria for feature id '20' Navigation have been satisfied.</b>
2.1	The navigation menu has a link to the 'Songbook' page that is styled correctly for the site.
3	<b>The following acceptance criteria for feature id '8' Songbook List have been satisfied.</b>
3.1	The songbook list page has the heading 'Songbook' just below the header.
3.2	The songbook list has an alphabetically ordered list of all the songs that a user has entered.
3.3	The view a list of songs page has the regular site header, footer and navigation bar with their regular associated links (as defined in the full landing page (id 1) and navigation (id 20) feature acceptance criteria lists)
3.4	The songbook list page is responsive and resizes for the screen on which it is being viewed.
3.5	The songbook list becomes scrollable if the list is too long to fit on the page.



user story id	user story name
	<b>3 View a song</b>
<b>Description</b>	As a Site User I can view a song I've entered and its associated details so that I can make reference to them as required.
<b>Story points</b>	3
<b>MoSCoW Category</b>	Must Have
<b>Epic.User story Dependencies</b>	2.2
Task id	Tasks
	<b>1</b> Update songbook list so that each song is a clickable link to the song ( template and view)
	<b>2</b> Code song view page template and view
	<b>3</b> Style the new front end features
	<b>4</b> Test the new song view functionality
Acceptance criteria id	Acceptance criteria
	<b>1</b> The user can view a song that they have entered along with all its associated data.
	<b>2</b> The following acceptance criteria for feature id '8' Songbook List have been satisfied.
	<b>2.1</b> Each song on the songbook list is a clickable link that takes the user to the songs view song page.
	<b>3</b> The following acceptance criteria for feature id '9' Song View have been satisfied.
	<b>3.1</b> The view song page shows the title of the song above the lyrics.
	<b>3.2</b> If the artist for the song was entered by the user it is displayed below the song title.
	<b>3.3</b> The song view page displays the lyrics for the selected song.
	<b>3.4</b> The lyrics on the song view page become scrollable if the lyrics are too long to fit on the screen.
	<b>3.5</b> The song view page has a 'songbook' button at the bottom that returns the user to the songbook list if clicked.
	<b>3.6</b> The 'songbook' button is fixed at the bottom of the screen above the footer.
	<b>3.7</b> The song view page and all its content are styled correctly for the site.
	<b>3.8</b> The view a song page has the regular site header, footer and navigation bar with their regular associated links (as defined in the full landing page (id 1) and navigation (id 20) feature acceptance criteria lists)
	<b>3.9</b> The song view page is responsive and resizes for the screen on which it is being viewed.

user story id	user story name
4	Edit a song
Description	As a Site User I can update a song I've entered so I can alter details as necessary.
Story points	1
MoSCoW Category	Must Have
Epic.User story Dependencies	2.3
Task id	Tasks
1	Update the song view page view and template as necessary
2	Update the add song page view and template and urls so reusable as song edit page
3	Style the new front end features
4	Test all new features
Acceptance criteria id	Acceptance criteria
1	The user can edit a song that they have entered.
2	<b>The following acceptance criteria for feature id '9' Song View have been satisfied.</b>
2.1	There is an 'Edit' button on the song view page, that takes the user to the song edit page on click.
3	<b>The following acceptance criteria for feature id '10' Edit song have been satisfied.</b>
3.1	The edit page has the heading 'Song' just below the header.
3.2	The edit song page has a title field that is validated and allows up to 150 characters.
3.3	The edit song page has an artist field that is validated and allows up to 150 characters.
3.4	The edit song page has a text box lyrics field that allows for characters to be entered.
3.5	The edit song page has a save button that saves the user entered data then returns the user to the songbook list page
3.6	The edit song page has the regular site header, footer and navigation bar with their regular associated links (as defined in the full feature acceptance criteria list)
3.7	The edit song page is styled correctly for the site.
3.8	The edit song page is responsive and resizes for the screen on which it is being viewed.
3.9	The edit song page has the regular site header, footer and navigation bar with their regular associated links (as defined in the full landing page (id 1) and navigation (id 20) feature acceptance criteria lists)

user story id	user story name
5	Delete a song
Description	As a Site User I can delete a song I've entered from songbook so it can no longer be accessed.
Story points	1
MoSCoW Category	Must Have
Epic.User story Dependencies	2.4
Task id	Tasks
1	Update the add song template to show a 'delete' button in place of the 'exit' button.
2	Code so that clicking delete button works as specified
3	Style the new front end features
4	Test new functionality
Acceptance criteria id	Acceptance criteria
1	The user can delete a song that they have entered.
2	<b>The following acceptance criteria for feature id '10' Edit song have been satisfied.</b>
2.1	The edit song page has a delete button that deletes the song currently shown in the edit song page and is styled consistently with the rest of the site.
3	<b>The following acceptance criteria for feature id '11' Delete song have been satisfied.</b>
3.1	On pressing the delete button, the user is asked to confirm that they wish to delete the song.
3.2	If the user states that they do not wish to delete the selected song, they are returned to the edit song page.
3.3	On successful deletion of the selected song, the user is redirected to the songbook list page.
3.4	The site user clicking that they do not wish to confirm deletion returns them to the previous page.

user story id	user story name
6	<b>Administer a song</b>
<b>Description</b>	As a Site Admin I can add, view, edit and delete songs of any user should corrective administrative action be necessary.
<b>Story points</b>	1
<b>MoSCoW Category</b>	Must Have
<b>Epic.User story Dependencies</b>	2.1
<b>Task id</b>	<b>Tasks</b>
1	Configure admin for song administration
2	Configure the admin page song edit page to use summernote for lyric create, view, edit and delete functions
3	Test the new admin features
<b>Acceptance criteria id</b>	<b>Acceptance criteria</b>
1	A logged in Site Admin can add, view, edit and delete songs of any user should corrective administrative action be necessary.
2	The following acceptance criteria for feature id '4' Admin Page have been satisfied.
2.1	There is a song admin section on the site admin page
2.2	All songs are listed on the song admin section of the site admin page
2.3	A site admin can add a song via the song admin section of the site admin page.
2.4	A site admin can select a song from the song list in the song admin section of the admin page and be taken to the admin edit song section of the site admin page.
2.5	A site admin can view a song and all its associated data on the admin edit song section of the admin page.
2.6	A site admin can edit a song and all its associated data on the admin edit song section of the admin page.
2.7	A site admin can select multiple songs from the song list in the song section of the site admin page and delete them
2.8	On clicking to delete multiple songs, the admin is asked to confirm that they'd like to delete them.
2.9	There is a delete button on the admin edit song section of the site admin page that deletes the selected song and all its associated data.
2.10	The site admin is asked to confirm that they would like to delete the song prior to the song deletion being actioned.
2.11	The site admin clicking that they do not wish to confirm deletion returns them to the previous page.
2.12	All songs belonging to the user are deleted when the associated user is deleted by a site admin via the user section of the site admin page.
2.13	The site admin page is responsive and resizes for the screen on which it is being viewed.

user story id	user story name
7	<b>Autoscroll a song</b>
<b>Description</b>	As a Site User viewing a song, I can initiate and stop autoscroll at the speed set when adding or editing a song so that I can sing or play along to the lyrics without having to manually scroll them.
<b>Story points</b>	5
<b>MoSCoW Category</b>	Could Have
<b>Epic.User story Dependencies</b>	2.3
Task id	Tasks
1	Update the song model to contain a play speed field.
2	Update the add / edit song page to enable the setting of a play speed
3	Update the song view page template and view as required
4	Code the autoscroll functionality in python and JavaScript
5	Style the new front end features
6	Test the new functionality
Acceptance criteria id	Acceptance criteria
1	A Site User viewing a song can initiate and stop autoscroll at the speed set when adding or editing a song so that they can sing or play along to the lyrics without having to manually scroll them.
2	<b>The following acceptance criteria for feature id '7' Add Song have been satisfied.</b>
2.1	The add song page has a play speed field that allows a speed of 1 - 5 to be selected.
3	<b>The following acceptance criteria for feature id '10' Edit song have been satisfied.</b>
3.1	The edit song page has a play speed field that allows a speed of 1 - 5 to be selected.
4	<b>The following acceptance criteria for feature id '9' Song View have been satisfied.</b>
4.1	The song view page has a play button just above the footer.
4.2	There is a label with the user set autoscroll speed that is positioned just above the footer on the song view page.
5	<b>The following acceptance criteria for feature id '13' Autoscroll have been satisfied.</b>
5.1	On click the play button is changed to an 'exit' button.
5.2	On clicking the play button, the lyrics stretch over the header and the footer, the exit button is displayed at the bottom of the page and autoscroll initiates.
5.3	The user is returned to the song view page if the 'exit' button is pressed during autoscroll

	<p>On the bottom of the lyrics being reached during autoscroll, the page waits until the user presses 'exit' button. (This is in case the musician is playing to the lyrics that are still displayed). When the user presses the exit button, the</p> <p><b>5.4</b> lyrics reduce in size revealing the site header footer.</p>
--	---

<b>Epic Id</b>	<b>Epic</b>
<b>3</b>	<b>Setlist</b>
<b>Description</b>	<b>Implement a way to group song lyrics</b>

<b>user story id</b>	<b>user story name</b>
<b>1</b>	<b>Add a set of songs</b>
<b>Description</b>	As a Site User I can add a set title and then select the songs I would like as part of that set.
<b>Story points</b>	5
<b>MoSCoW</b>	
<b>Category</b>	Should Have
<b>Epic.User story Dependencies</b>	3.2
<b>Task id</b>	<b>Tasks</b>
<b>1</b>	Update database schema to contain set name, songs in set, slug and artist
<b>2</b>	Update the setlist list page template and view as necessary
<b>3</b>	Code the add setlist view, template and urls
<b>4</b>	Style the new front end features
<b>5</b>	Test the new functionality
<b>Acceptance criteria id</b>	<b>Acceptance criteria</b>
<b>1</b>	A Site User can add a set title and then select the songs they would like as part of that set.
<b>2</b>	<b>The following acceptance criteria for feature id '14' Setlist List have been satisfied.</b>
<b>2.1</b>	The setlist list page has an 'Add' button that navigates to the add setlist page on click.
<b>2.2</b>	The 'Add' button is styled consistently with the rest of the site.
<b>3</b>	<b>The following acceptance criteria for feature id '16' Add Setlist have been satisfied.</b>
<b>3.1</b>	The add setlist page has the heading 'Setlist' at the top just below the header that is styled consistently with the rest of the site.
<b>3.2</b>	The add setlist page has a set name field that is validated, is required and allows up to 150 characters.
<b>3.3</b>	The add setlist page shows a list of all the songs entered by the user with unchecked checkboxes next to them to enable the user to select the songs desired as part of the new set.
<b>3.4</b>	The add setlist page has a 'save' button that saves the new setlist and returns the user to the setlist list page when clicked.
<b>3.5</b>	The save button is styled consistently with the rest of the site.
<b>3.6</b>	The add setlist page has an 'Exit' button that returns the user to the setlist list page when clicked.
<b>3.7</b>	The exit button is styled consistently with the rest of the site.

3.8	The add setlist page has the regular site header, footer and navigation bar with their regular associated links (as defined in the full landing page (id 1) and navigation (id 20) feature acceptance criteria lists)
3.9	The add setlist page is responsive and resizes for the screen on which it is being viewed.



user story id	user story name
2	<b>View a list of sets</b>
Description	As a Site User I can view a list of all the sets I've created so that I can select the one I want to access.
Story points	3
MoSCoW Category	Should Have
Epic.User story Dependencies	2.1
Task id	Tasks
1	Update navigation as required
2	Code the setlist list page view, template and urls
3	Style the new front end features
4	Test the new functionality
Acceptance criteria id	Acceptance criteria
1	A Site User can view a list of all the sets they've created so that they can select the one they want to access.
2	<b>The following acceptance criteria for feature id '20' Navigation have been satisfied.</b>
2.1	The navigation menu has a link to 'setlist'
2.2	The link to setlist is styled correctly for the site.
3	<b>The following acceptance criteria for feature id '14' Setlist list have been satisfied.</b>
3.1	The setlist list page has the heading 'Setlists' at the top just below the header that is styled consistently with the rest of the site.
3.2	The setlist list page show a list of all the setlists that the user has created.
3.3	The setlist list becomes scrollable should the list become too long to fit on the page.
3.4	The setlist list page is responsive and resizes for the screen on which it is being viewed.
3.5	The view a list of sets page has the regular site header, footer and navigation bar with their regular associated links (as defined in the full landing page (id 1) and navigation (id 20) feature acceptance criteria lists)

user story id	user story name
3	View a set of songs
Description	As a Site User I can view a list of all the songs in a selected set so I don't have to look through my entire list of songs for the ones I need at that time.
Story points	3
MoSCoW Category	Should Have
Epic.User story Dependencies	3.1
Task id	Tasks
1	Code the setlist song list page template view and urls
2	Update the setlist list page so that the setlist names in the list are clickable links to a setlist songlist page
3	Style the new front end features
4	Test the new functionality
Acceptance criteria id	Acceptance criteria
1	A Site User can view a list of all the songs in a selected set so they don't have to look through their entire list of songs for the ones they need at that time.
2	Deletion of a song that is linked to a set or sets does not cause an error when trying to view a set of songs that used to contain that song
3	<b>The following acceptance criteria for feature id '14' Setlist list have been satisfied.</b>
3.1	Each set in the setlist list is a clickable link that takes the user to the setlist view page displaying the contents of the clicked set.
4	<b>The following acceptance criteria for feature id '15' Setlist song list have been satisfied.</b>
4.1	The setlist song list page displays the set name just below the header and it is styled consistently with the rest of the site.
4.2	The setlist song list page displays a list of all the songs selected to be part of that set
4.3	The setlist list page has a 'Setlists' button at the bottom just above the footer that returns the user to the list of sets page when clicked. The button is styled consistently with the rest of the site.
4.4	The view a set of songs page has the regular site header, footer and navigation bar with their regular associated links (as defined in the full landing page (id 1) and navigation (id 20) feature acceptance criteria lists)
4.5	The setlist song list page is responsive and resizes for the screen on which it is being viewed.

user story id	user story name
	<b>4 View a song from a set</b>
Description	As a Site User I can view a song from a set and its associated details so that I can make reference to them as required.
Story points	1
MoSCoW Category	Should Have
Epic.User story Dependencies	3.3
Task id	Tasks
1	Update song view template and views so that it can also be used for setlist song view
2	Update setlist song list so that each song in the list is a clickable link to the setlist song view page for the clicked song
3	Style the new front end features
4	Test the new functionality
Acceptance criteria id	Acceptance criteria
1	A Site User can view a song from a set and its associated details so that they can make reference to them as required.
2	<b>The following acceptance criteria for feature id '15' Setlist song list have been satisfied.</b>
2.1	Each song in the setlist song list is a clickable link that takes the user to the setlist song view page displaying the contents of the clicked song.
3	<b>The following acceptance criteria for feature id '21' Setlist Song View have been satisfied.</b>
3.1	The setlist view song page shows the title of the song above the lyrics.
3.2	If the artist for the song was entered by the user it is displayed below the song title.
3.3	The setlist song view page displays the lyrics for the selected song.
3.4	The lyrics on the setlist song view page become scrollable if the lyrics are too long to fit on the screen.
3.5	The setlist song view page has a 'Setlist' button at the bottom returns the user to the Setlist song list if clicked.
3.6	The 'Setlist' button is fixed at the bottom of the screen above the footer.
3.7	The setlist song view page has an 'Edit' button at the bottom just above the footer that takes the user to the Edit song page when clicked.
3.8	The setlist song view page and all its content are styled correctly for the site.
3.9	The view a song from a set page has the regular site header, footer and navigation bar with their regular associated links (as defined in the full landing page (id 1) and navigation (id 20) feature acceptance criteria lists)

<b>3.10</b>	The setlist song view page is responsive and resizes for the screen on which it is being viewed.
-------------	--

user story id	user story name
5	<b>Edit a Set</b>
<b>Description</b>	As a Site User I can update a set of songs so I can change the songs in that set as necessary.
<b>Story points</b>	1
<b>MoSCoW Category</b>	Should Have
<b>Epic.User story Dependencies</b>	3.2
Task id	Tasks
1	Update the add set page template and view so that it can also be used as the edit set page
2	Update the setlist song list page so that it has an edit button at the bottom
3	Style the new front end features
4	Test the new functionality
Acceptance criteria id	Acceptance criteria
1	A Site User can update a set of songs so they can change the songs in that set as necessary.
2	<b>The following acceptance criteria for feature id '15' Setlist Song list have been satisfied.</b>
2.1	The setlist song list page has an 'Edit' button at the bottom of the page just above the footer that will take the user to the edit set page when clicked. The 'Edit' button is styled consistently with the rest of the site.
3	<b>The following acceptance criteria for feature id '17' Edit Set have been satisfied.</b>
3.1	The edit setlist page has the heading 'Setlist' at the top just below the header that is styled consistently with the rest of the site.
3.2	The edit setlist page has a set name field that is validated, is required and allows up to 150 characters.
3.3	The edit setlist page shows a list of all the songs entered by the user with checkboxes next to them to enable the user to select the songs desired as part of the set. The checkboxes are pre ticked with songs currently in the set.
3.4	The edit setlist page has a 'save' button that saves the new setlist and returns the user to the setlist list page when clicked.
3.5	The save button is styled consistently with the rest of the site.
3.6	The edit a set page has the regular site header, footer and navigation bar with their regular associated links (as defined in the full landing page (id 1) and navigation (id 20) feature acceptance criteria lists)
3.7	The edit setlist page is responsive and resizes for the screen on which it is being viewed.

user story id	user story name
6	Delete a set
Description	As a Site User I can delete a set of songs so it can no longer be accessed.
Story points	1
MoSCoW Category	Should Have
Epic.User story Dependencies	3.5
Task id	Tasks
1	Code the delete set functionality
2	Update the edit set page so that it has a delete button
3	Style the new front end features
4	Test the new functionality
Acceptance criteria id	Acceptance criteria
1	A Site User can delete a set of songs so it can no longer be accessed.
2	<b>The following acceptance criteria for feature id '17' Edit Set have been satisfied.</b>
2.1	The edit setlist page has a 'Delete' button styled consistently with the rest of the site.
3	<b>The following acceptance criteria for feature id '18' Delete Set have been satisfied.</b>
3.1	The user is asked to confirm that they wish to delete the selected set.
3.2	If the user states that they do not wish to delete the selected set, they are returned to the edit set page.
3.3	On confirming that deletion is required, the deletion takes place and user is returned to the setlist list page.

user story id	user story name
7	<b>Administer a set of songs</b>
Description	As a site Admin I can add, view, edit and delete sets of songs of any user should corrective administrative action be necessary.
Story points	1
MoSCoW Category	Should Have
Epic.User story Dependencies	
Task id	Tasks
1	Configure the admin page for set administration
2	Test the new functionality
Acceptance criteria id	Acceptance criteria
1	A site Admin can add, view, edit and delete sets of songs of any user should corrective administrative action be necessary.
2	<b>The following acceptance criteria for feature id '4' Admin Page have been satisfied.</b>
2.1	There is a set admin section on the site admin page
2.2	All sets are listed on the set admin section of the site admin page
2.3	A site admin can add a set via the set admin section of the site admin page.
2.4	A site admin can select a set from the set list in the set admin section of the admin page and be taken to the admin edit set section of the site admin page.
2.5	A site admin can view a set and all its associated data on the admin edit set section of the admin page.
2.6	A site admin can edit a set and all its associated data on the admin edit set section of the admin page.
2.7	A site admin can select multiple sets from the set list in the set section of the site admin page and delete them
2.8	On clicking to delete multiple sets, the admin is asked to confirm that they'd like to delete them.
2.9	The site admin clicking that they do not wish to confirm deletion of multiple sets returns them to the setlist page in site admin
2.10	There is a delete button on the admin edit set section of the site admin page that deletes the selected set and its title
2.11	The site admin is asked to confirm that they would like to delete the set prior to the set deletion being actioned.
2.12	The site admin clicking that they do not wish to confirm deletion of a given set returns them to the edit set page for that set
2.13	All sets belonging to the user are deleted when the associated user is deleted by a site admin via the user section of the site admin page.
2.14	The site admin setlist admin page is responsive and resizes for the screen on which it is being viewed.

user story id	user story name
	<b>8 Autoscroll a setlist song</b>
<b>Description</b>	As a site Admin I can add, view, edit and delete sets of songs of any user should corrective administrative action be necessary.
<b>Story points</b>	5
<b>MoSCoW Category</b>	Could Have
<b>Epic.User story Dependencies</b>	3.4
Task id	Tasks
1	Update setlist song view to have autoscroll functionality
2	Test the new functionality
Acceptance criteria id	Acceptance criteria
1	A Site User viewing a setlist song can initiate and stop autoscroll at the speed set when adding or editing a song so that they can sing or play along to the lyrics without having to manually scroll them.
	<b>The following acceptance criteria for feature id '21' Setlist Song View have been satisfied.</b>
2.1	The setlist song view page has a play button just above the footer.
2.2	There is a label with the user set autoscroll speed that is positioned just above the footer on the setlist song view page.
	<b>The following acceptance criteria for feature id '22' Setlist Song Autoscroll have been satisfied.</b>
3.1	On click the play button is changed to an 'exit' button.
3.2	On clicking the play button, the lyrics stretch over the header and the footer, the exit button is displayed at the bottom of the page and autoscroll initiates.
3.3	The user is returned to the setlist song view page if the 'exit' button is pressed during autoscroll
3.4	On the bottom of the lyrics being reached during autoscroll, the page waits until the user presses 'exit' button. (This is in case the musician is playing to the lyrics that are still displayed). When the user presses the exit button, the lyrics reduce in size revealing the site header footer.