AMRITA VISHWA VIDYAPEETHAM Academic Year 2020-21

Semester: VI Sem B.Tech 15CSE313 SOFTWARE ENGINEERING

Backlog Grooming

Team Name: Pipeline Predators
Project Title: QUIZ APP

Roll Number	Name	Role for Sprint 1	Marks (10)
CB.EN.U4CSE18417	Dharsan R.	Lead Programmer	
CB.EN.U4CSE18425	Hayma Sunder P.	Project Owner	
CB.EN.U4CSE18430	Kiran Kumar A.	Scrum Master	
CB.EN.U4CSE18433	Phanindra Reddy K	User Experience Designer	
CB.EN.U4CSE18464	Sandhya V	Frontline Tester	

Faculty name and signature:

Epic 1 Student/Faculty Login

T-Shirt Size: S

Story 1 – Story Points - 3
As a user, I must be able to view the initial Home-Screen which has provisions to enter credentials.

Task	Member	Priority	Estimate (Hours)
Task 1	Kiran Kumar	M	3
Task 2	Phanindra Reddy	C	4
Task 3	Sandhya	M	2
Task 4	Hayma Sunder	M	1
Task 5	Dharsan	S	1
Task 6	Hayma Sunder	C	2

Story 2: Story Points - 3
As an app-user, I must be provided with the provision for Multi-User login: Faculty or Student.

Task	Member	Priority	Estimate (Hours)
Task 1	Phanindra Reddy	M	2
Task 2	Kiran Kumar	S	1
Task 3	Kiran Kumar	M	1
Task 4	Sandhya	S	3
Task 5	Hayma Sunder	M	2
Task 6	Dharsan	M	3

Story 3: Story Points - 2
Comparing input credentials against the identity stored in the backend database.

Task	Member	Priority	Estimate (Hours)
Task 1	Phanindra Reddy	M	2
Task 2	Hayma Sunder	M	1.5
Task 3	Kiran Kumar	M	1

Task 4	Sandhya	S	1
Task 5	Dharsan	S	1
Task 6	Phanindra Reddy	S	1

Story 4: Story Points - 2
As an app-user, I must be redirected to the next page based on my credibility.

Task	Member	Priority	Estimate (Hours)
Task 1	Kiran Kumar	M	2
Task 2	Sandhya	M	0.5
Task 3	Hayma Sunder	S	0.5
Task 4	Dharsan	M	1
Task 5	Phanindra Reddy	C	0.5
Task 6	Sandhya	C	0.5

Story 5: Story Points - 2
Forgot Password Mechanism.

Task	Member	Priority	Estimate (Hours)
Task 1	Member 1	M	2
Task 2	Kiran Kumar	S	1.5
Task 3	Phanindra Reddy	M	1.5
Task 4	Hayma Sunder	M	1
Task 5	Sandhya	M	1
Task 6	Dharsan	M	0.5

Epic 2 Dashboard Design

T-Shirt Size: L

Story 1: Story Points - 4
As a student, I must be able to see all the courses I registered for and notifications for any quizzes scheduled for the day.

Task	Member	Priority	Estimate (Hours)
Task 1	Dharsan	M	3
Task 2	Phanindra Reddy	M	2
Task 3	Hayma Sunder	C	3
Task 4	Sandhya	C	2
Task 5	Kiran Kumar	M	2
Task 6	Hayma Sunder	S	1

Story 2: Story Points - 5
As a student, I must be able to see all the quizzes attended in a particular course.

Task	Member	Priority	Estimate (Hours)
Task 1	Hayma Sunder	M	2
Task 2	Dharsan	C	3
Task 3	Kiran Kumar	M	3
Task 4	Phanindra Reddy	W	3
Task 5	Kiran Kumar	S	3
Task 6	Sandhya	S	3.5

Story 3: Story Points - 4
As a faculty, I should be able to see the list of courses handled by me and ongoing quizzes scheduled by me.

Task	Member	Priority	Estimate (Hours)
Task 1	Hayma Sunder	M	2
Task 2	Kiran Kumar	S	2
Task 3	Sandhya	M	3
Task 4	Phanindra Reddy	C	2

Task 5	Dharsan	W	3
Task 6	Phanindra Reddy	S	2

Story 4: Story Points - 4

As a faculty, I must be able to schedule a quiz or see all the quizzes conducted in that course.

Task	Member	Priority	Estimate (Hours)
Task 1	Kiran Kumar	M	2
Task 2	Sandhya	S	3
Task 3	Sandhya	M	2
Task 4	Dharsan	S	3
Task 5	Hayma Sunder	S	3
Task 6	Phanindra Reddy	C	2

Story 5: Story Points - 3

Personal information can be viewed and edited in the ABOUT section.

Task	Member	Priority	Estimate (Hours)
Task 1	Kiran Kumar	C	3
Task 2	Phanindra Reddy	C	3
Task 3	Dharsan	C	1.5
Task 4	Phanindra Reddy	C	2
Task 5	Sandhya	C	1
Task 6	Hayma Sunder	C	1

Epic 3 Quiz Management

T-Shirt Size: M

Story 1: Story Points - 4

As a faculty, I must be able to choose an appropriate time and duration for the quiz.

Task	Member	Priority	Estimate (Hours)
Task 1	Sandhya	S	5
Task 2	Hayma Sunder	S	3
Task 3	Kiran Kumar	S	2
Task 4	Dharsan	C	0.5
Task 5	Phanindra Reddy	S	1
Task 6	Kiran Kumar	C	2

Story 2: Story Points - 5

As a Quiz-Setter, I must be able to set-up questions for a quiz.

Task	Member	Priority	Estimate (Hours)
Task 1	Hayma Sunder	M	5
Task 2	Kiran Kumar	M	2
Task 3	Phanindra Reddy	C	3
Task 4	Dharsan	S	3
Task 5	Sandhya	S	3
Task 6	Hayma Sunder	M	3

Story 3: Story Points - 3

As a Quiz-Setter, I can either set/edit the password and release the time of the quiz.

Task	Member	Priority	Estimate (Hours)
Task 1	Hayma Sunder	C	3
Task 2	Phanindra Reddy	C	1.5
Task 3	Sandhya	W	2

Task 4	Kiran Kumar	W	2
Task 5	Phanindra Reddy	W	1.5
Task 6	Dharsan	C	2

Story 4: Story Points - 3

As a student, I must be able to access the quiz at the release time.

Task	Member	Priority	Estimate (Hours)
Task 1	Phanindra Reddy	C	3
Task 2	Kiran Kumar	C	2
Task 3	Hayma Sunder	C	2
Task 4	Kiran Kumar	S	1
Task 5	Sandhya	S	1.5
Task 6	Dharsan	C	1.5

Story 5: Story Points - 2

As a faculty, I can manage each individual quiz-takers feedback right after they submit.

Task	Member	Priority	Estimate (Hours)
Task 1	Kiran Kumar	W	2
Task 2	Sandhya	W	1
Task 3	Phanindra Reddy	W	1
Task 4	Dharsan	W	1
Task 5	Phanindra Reddy	W	1
Task 6	Hayma Sunder	W	1.5

Epic 4 Taking up the quiz

T-Shirt Size: M

Story 1: Story Points - 4

As a faculty, I should have access to the ordering of the questions.

Task	Member	Priority	Estimate (Hours)
Task 1	Kiran Kumar	C	3
Task 2	Phanindra Reddy	C	2
Task 3	Kiran Kumar	C	3
Task 4	Sandhya	C	2
Task 5	Hayma Sunder	C	2
Task 6	Dharsan	C	3

Story 2: Story Points - 3

As a Student, I must be able to start the quiz and navigate through the questions.

Task	Member	Priority	Estimate (Hours)
Task 1	Hayma Sunder	M	2
Task 2	Sandhya	C	2
Task 3	Phanindra Reddy	M	1.5
Task 4	Dharsan	C	1.5
Task 5	Kiran Kumar	S	1.5
Task 6	Sandhya	M	2

Story 3: Story Points - 3

As a Quiz taker, I should be able to view the details of each question.

Task	Member	Priority	Estimate (Hours)
Task 1	Hayma Sunder	C	1.5

Task 2	Phanindra Reddy	M	2
Task 3	Hayma Sunder	M	2
Task 4	Kiran Kumar	M	1
Task 5	Dharsan	M	2
Task 6	Sandhya	M	1.5

Story 4: Story Points - 3

As a Quiz taker, I want to monitor the timer during my course of the test.

Task	Member	Priority	Estimate (Hours)
Task 1	Dharsan	S	2
Task 2	Hayma Sunder	S	2
Task 3	Sandhya	C	1.5
Task 4	Kiran Kumar	S	3
Task 5	Phanindra Reddy	S	1.5
Task 6	Phanindra Reddy	C	1.5

Story 5: Story Points - 4

As a faculty, I want the performance analysis for each student for the quiz.

Task	Member	Priority	Estimate (Hours)
Task 1	Kiran Kumar	M	3
Task 2	Dharsan	S	3
Task 3	Phanindra Reddy	M	2.5
Task 4	Dharsan	C	1.5
Task 5	Sandhya	C	3
Task 6	Hayma Sunder	C	1.5

Epic 5 Results Page

T-Shirt Size: L

Story 1: Story Points - 4

Once the quiz gets over the result screen must be shown.

Task	Member	Priority	Estimate (Hours)
Task 1	Dharsan	M	3
Task 2	Phanindra Reddy	M	2
Task 3	Sandhya	C	3
Task 4	Hayma Sunder	M	1
Task 5	Kiran Kumar	M	2
Task 6	Dharsan	M	2

Story 2: Story Points - 4

As a student, I must be able to view my score and correct answers after completing the test.

Task	Member	Priority	Estimate (Hours)
Task 1	Phanindra Reddy	M	4
Task 2	Kiran Kumar	M	2
Task 3	Hayma Sunder	S	2
Task 4	Sandhya	S	1.5
Task 5	Dharsan	C	2
Task 6	Kiran Kumar	S	2

Story 3: Story Points - 4

The performance of the students must be shown to all stakeholders.

Task	Member	Priority	Estimate (Hours)
Task 1	Dharsan	C	4
Task 2	Hayma Sunder	M	2
Task 3	Phanindra Reddy	W	2

Task 4	Sandhya	C	3
Task 5	Hayma Sunder	W	1
Task 6	Kiran Kumar	C	1

Story 4: Story Points - 3

As a faculty, I must be able to analyse the results after conducting a quiz.

Task	Member	Priority	Estimate (Hours)
Task 1	Sandhya	C	2
Task 2	Sandhya	C	2
Task 3	Dharsan	C	2
Task 4	Phanindra Reddy	C	1.5
Task 5	Kiran Kumar	S	1.5
Task 6	Hayma Sunder	S	3

Story 5: Story Points - 3

A leader-board must be present honouring the toppers of that specific test.

Task	Member	Priority	Estimate (Hours)
Task 1	Kiran Kumar	C	2
Task 2	Hayma Sunder	C	3
Task 3	Sandhya	C	1.5
Task 4	Dharsan	C	1.5
Task 5	Phanindra Reddy	C	1.5
Task 6	Dharsan	W	2

Epic 6 Miscellaneous

T-Shirt Size: S

Story 1: Story Points - 2

Passwords of users are encrypted and stored in the database.

Task	Member	Priority	Estimate (Hours)
Task 1	Dharsan	W	1
Task 2	Kiran Kumar	W	1
Task 3	Phanindra Reddy	W	0.5
Task 4	Sandhya	W	0.5
Task 5	Hayma Sunder	W	0.5
Task 6	Dharsan	W	1

Story 2: Story Points - 2

As an admin of the application, I should be able to access and edit everything.

Task	Member	Priority	Estimate (Hours)
Task 1	Hayma Sunder	C	1
Task 2	Phanindra Reddy	W	1
Task 3	Dharsan	C	3
Task 4	Kiran Kumar	C	1
Task 5	Sandhya	C	0.5
Task 6	Hayma Sunder	C	1

Story 3: Story Points - 2

Pushing Notifications.

Task	Member	Priority	Estimate (Hours)
Task 1	Dharsan	C	2
Task 2	Kiran Kumar	C	2
Task 3	Phanindra Reddy	C	1
Task 4	Sandhya	C	1

Task 5	Hayma Sunder	C	0.5
Task 6	Sandhya	C	0.5

Story 4: Story Points - 3

As a user, I like to have an option for dark mode.

Task	**Member**	**Priority**	**Estimate (Hours)**
Task 1	Hayma Sunder	W	1
Task 2	Sandhya	W	2
Task 3	Phanindra Reddy	W	1
Task 4	Kiran Kumar	W	1
Task 5	Dharsan	W	1.5
Task 6	Dharsan	W	2

Story 5: Story Points - 3

Developers homage and reporting a bug.

Task	**Member**	**Priority**	**Estimate (Hours)**
Task 1	Phanindra Reddy	W	3
Task 2	Hayma Sunder	W	0.5
Task 3	Kiran Kumar	W	2
Task 4	Sandhya	W	2
Task 5	Sandhya	W	1
Task 6	Dharsan	W	1

****

Hayma Sunder P. – Project Owner (CSE18425)

Kiran Kumar A. – Srcum Master (CSE18430)

Dharsan R. – Lead Programmer (CSE18417)

Sandhya V. – Frontline Tester (CSE18464)

Phanindra Reddy K. – User Experience Designer

EPIC 1 – STUDENT / FACULTY LOGIN

Story 1:
As a user, I must be able to view the initial Home-Screen which has provisions to enter credentials.

Task 1: Write HTML code to display credential entry box.
Task 2: Integrate CSS to beautify and rightly place display elements.
Task 3: Provide an option (radio-button) to choose the type of user login student/faculty.
Task 4: Design text fields to record user-input.
Task 5: Integrate placeholders for the text boxes with appropriate names.
Task 6: Display hyperlinks for forgotten password mechanisms.

Story 2:
As an app-user, I must be provided with the provision for Multi-User login: Faculty or Student.

Task 1: Render individual options for multi-faceted user login.
Task 2: Restraining the user to select only either of the options and not both.
Task 3: Making it mandatory to select the type of user login.
Task 4: Issuing a prompt if required information is not entered.
Task 5: Recording the input, to track the frequency of user access.
Task 6: Forwarding the recorded input to the next phase - comparison.

Story 3:
Comparing input credentials against the identity stored in the backend database.

Task 1: Parsing the input credentials separately using POST HTTP requests to the database.
Task 2: De-encapsulating the encrypted user-credentials.
Task 3: Identity Matching performed individually on the password + roll number combination.

Task 4: Upon Success notify the user with a green signal and proceed to render the next page.
Task 5: Upon Repudiation, notify the user with an appropriate error message.
Task 6: Clear the recorded input data from the text field, prompting the user for another entry.

Story 4:
As an app-user, I must be redirected to the next page based on my credibility.

Task 1: Start to process rendering the next page if the identity of the user is credible.
Task 2: Render the student page/faculty page depending on the type of login ticked.
Task 3: Reload the page in case of any on-the-go mishap.
Task 4: Make sure the right page is loaded onto the user's browser and not the contrary.
Task 5: Prompt the user to manually reload the page if the loading problem persists.
Task 6: Prompt the user to re-enter the credentials if the loading problem persists.

Story 5:
Forgot Password Mechanism.

Task 1: Produce a new dialog box/web page if the user clicks on the forgot password option.
Task 2: Ask for user email to re-confirm whether the user actually requested a password reset.
Task 3: Send an email containing a confirm password reset button.
Task 4: If the button is clicked, the user is allowed to enter his new password twice for confirmation.
Task 5: The new password is updated in the database, in place of an already existing one.
Task 6: Now on entering the changed password, the user must be authorized to access the application.

Testing Tasks:

Task 1: Check if all the elements on the page are correctly placed for different browsers.
Task 2: Check if the page is responsive on different devices.
Task 3: Check if the password is correctly matched against the username in authentication for both faculty and student.
Task 4: Check if the user is redirected appropriately after authentication.
Task 5: Check if the confirmation mail is correctly sent after clicking the forgot password and check if the details are updated in the database by trying to login with the new password.
Task 6: Testing the API with software like postman.

Development Tasks:

Task 1: Create login table in database.
Task 2: Create API calls to read from database.
Task 3: Code React functions for authentication.
Task 4: Code for the responsive UI.
Task 5: Create function for forgotten password mechanism**.**
Task 6: Write function for auto login function.

EPIC 2 – DASHBOARD DESIGN

Story 1:
As a student, I must be able to see all the courses I registered for and notifications for any quizzes scheduled for the day.

Task 1: Get details of all the courses registered by the student for the semester.
Task 2: Write HTML code to display the notification box and list of courses for the semester.
Task 3: Integrate CSS to beautify and rightly place the display elements.
Task 4: Get the quiz details (scheduled for the day) from the database and display them in the notification box.
Task 5: If the quiz link is clicked, it should navigate to the quiz page.
Task 6: If a course is clicked, the user is redirected to the page containing details of that course.

Story 2:
As a student, I must be able to see all the quizzes attended in a particular course.

Task 1: Write HTML code to display the course details and list of quizzes.
Task 2: Integrate CSS stylings to put the elements in place.
Task 3: Get the details of the particular course from the database and display them.
Task 4: Display a general statement about the student’s performance in that course.
Task 5: Get the list of quizzes in that course from the database and display them.
Task 6: If a quiz from the list is clicked, the user should be redirected to the result page of the quiz which contains detailed information about it.

Story 3:
As a faculty, I should be able to see the list of courses handled by me and ongoing quizzes scheduled by me.

Task 1: Get the list of all courses handled by the faculty from the database.
Task 2: Get the details of ongoing quizzes that the faculty has scheduled.
Task 3: Design and code the UI to display all the courses handled by the faculty and ongoing quizzes.
Task 4: Integrate the CSS styling to enhance the look of the page.
Task 5: If the user clicks on an ongoing quiz, he can see how many students are currently taking the quiz.
Task 6: If a course is clicked, it should navigate to the course page.

Story 4:
As a faculty, I must be able to schedule a quiz or see all the quizzes conducted in that course.

Task 1: Get the course details from the database and display them.

Task 2: Display two options to either schedule a quiz or see the list of quizzes already conducted.
Task 3: If the first option is chosen, navigate the faculty to the quiz setter page.
Task 4: Get the details of all quizzes conducted in the course so far from the database.
Task 5: Display the list of quizzes if the second option is chosen.
Task 6: If a quiz is selected, redirect to the page with details about the quiz.

Story 5: Personal information can be viewed and edited in the ABOUT section.

Task 1: Get the photo, Reg.No. and other personal details of the user from the database.
Task 2: Design and code the UI with the photo and basic information of the user.
Task 3: Display a button 'Update Profile' below the details of the user.
Task 4: If the button is clicked, the user should be able to modify his personal details.
Task 5: Display a button 'Save Changes' at the end of the update page.
Task 6: If the button is clicked, the changes made should be updated in the corresponding record in the database.

Testing Tasks:

Task 1: Check if the page is responsive on different devices.
Task 2: Check if the course details and quiz details are correctly acquired from the database and shown on the screen.
Task 3: Check if the real-time notifications for the quiz are displayed and removed on time.
Task 4: Check if the profile is correctly acquired from the database and the updated details (if any) are rightly updated to the database.
Task 5: Testing the API with software like postman.

EPIC 3 – QUIZ MANAGEMENT

Story 1:
As a faculty, I must be able to choose an appropriate time and duration for the quiz.

Task 1: Set-up a Calendar Structure enabling the faculty to select the quiz date.
Task 2: Set-up a Clock Structure enabling the faculty to select the quiz time.
Task 3: Input the duration of the quiz.
Task 4: Validate the quiz configuration and throw an error message appropriately.
Task 5: Update the fixture in the database.
Task 6: Notify conflicts in fixtures to corresponding Quiz-Setter.

Story 2:
As a Quiz-Setter, I must be able to set-up questions for a quiz.

Task 1: Input all questions and answers for the quiz
Task 2: Allocate maximum scorable mark for each question.
Task 3: Map each question with its respective course outcome.
Task 4: Provide a reasonable number of options for the tagged question.
Task 5: Include Radio/Check List buttons depending on the MCQ type.
Task 6: Create a new quiz and update all inputs in the database
Task 7: Give provisions to choose the access type of the quiz.
Task 8: Provide Previous/Next options based on access type.

Story 3:
As a Quiz-Setter, I can either set/edit the password and release the time of the quiz.

Task 1: Option to associate the password with the registered quiz.
Task 2: Display added instructions before the students take up the quiz.
Task 3: Provision to enable the setter to modify the scheduled quiz date before the quiz commences.
Task 4: Provision to enable the setter to modify scheduled quiz time before the quiz commences.
Task 5: Provision to enable the setter to modify scheduled quiz duration before the quiz commences.
Task 6: After setting up all configurations, Release the button to update the students' notification with the scheduled quiz.

Story 4:
As a student, I must be able to access the quiz at the release time.

Task 1: Push notification to student inbox regarding scheduled pending quizzes.
Task 2: Notify the password through email, seconds before the quiz commences.
Task 3: Remainder action to update the user regarding an upcoming quiz.
Task 4: Enter the correct password in the tab option given.

Task 5: Password is verified, in case of an incorrect password, a warning message is thrown else display the first question.
Task 6: Remove notification from the inbox as the student completes the quiz.

Story 5:
As a faculty, I can manage each individual quiz-takers feedback right after they submit.

Task 1: Ask feedback questions about technical issues faced during the quiz and record the response.
Task 2: A text field is made available for students to enter verbal feedback.
Task 3: Create a table in the database for recording the feedback received from the students.
Task 4: Populate the table with received student feedback responses.
Task 5: Faculty can access the feedback for the quiz if they have created that quiz.
Task 6: Provision to broadcast messages or reply to a particular student regarding the problem redressal.

Testing Tasks:

Task 1: Check if the calendar and clock structures to set up the quiz date and time are working properly.
Task 2: Check if the conflicts are correctly identified and intimated while scheduling a quiz.
Task 3: Check if all the kinds of settings are working in a quiz-setter (like input type, options, maximum marks, CO mapping, access type, etc.)
Task 4: Check if the faculty can set up and update the password, time, date, and duration of the quiz.
Task 5: Check if the notifications, password mails are sent on time to all the students.
Task 6: Check if the feedback form is displayed after each quiz and the collected information is updated to the database.
Task 7: Testing the API with software like postman.

EPIC 4 – TAKING UP THE QUIZ

Story 1:
As a faculty, I should have access to the ordering of the questions.

Task 1: The order of the questions for each student must be random.
Task 2: The order of the choices for each question must be random for each student.
Task 3: Validating the order of choices for each question.
Task 4: Verify that no question is repeated for each student.
Task 5: Maintain and update the order of questions for each student in the database.
Task 6: Ensure that the proper answer key is associated with each question and doesn't get affected due to the randomized order of questions.

Story 2:
As a Student, I must be able to start the quiz and navigate through the questions.

Task 1: Redirect to the quiz page once the quiz gets started.
Task 2: Provision to save the option for each question.
Task 3: Provision to go to the next question by clicking the next button.
Task 4: Provision to go to the previous question by clicking the previous button if and only if non-linear access for the quiz is enabled.
Task 5: Features to submit the quiz in case if the student completes the quiz early.
Task 6: Once the quiz is completed or the due time for the quiz is up, the quiz must be saved properly for each student and must get updated in the database.

Story 3:
As a Quiz taker, I should be able to view the details of each question.

Task 1: View the course outcome for each question.
Task 2: Question numbers and questions must be mapped correctly and displayed properly.
Task 3: Ensure special characters such as mathematical symbols are displayed correctly.
Task 4: Choose a single option, in case of a single answer type question with radio buttons.
Task 5: Select multiple choices, in case of a multiple answer type question with checklist buttons.
Task 6: Images are displayed if any as set by the faculty.

Story 4:
As a Quiz taker, I want to monitor the timer during my course of the test.

Task 1: The timer must be enabled as soon as the quiz is started by the student.
Task 2: Update the timer every second for each student.

Task 3: Notify the student about the time remaining during the last 5 minutes of the quiz.
Task 4: In case, the student logs out by mistake, the timer must freeze for the student until he logs back in.
Task 5: Once the duration of the quiz is over or the due time for the quiz is up, the quiz must close automatically.
Task 6: After the completion of the quiz, the total time taken by the student should get updated in the Database.

Story 5:
As a faculty, I want the performance analysis for each student for the quiz.

Task 1: Calculate the score for each student based on his performance.
Task 2: Allocate grades to each student based on his performance and determine if he passes or fails.
Task 3: Update the score and performance of each student in the database.
Task 4: Sort the students based on their marks and calculate the ranks of the student.
Task 5: Analyse the overall performance of the quiz.
Task 6: Update the analysis for the quiz in the database records.

Testing Tasks:

Task 1: Check if the performance metrics are working error-free.
Task 2: Check if iterating through questions in the quiz is working perfectly.
Task 3: Check if the timer is working correctly and the timer notification is working fine.
Task 4: Check how the random question is working in real time.
Task 5: Testing the API with software like postman.
Task 6: Check if radio buttons and checklists are working fine for mapped questions.
Task 7: Check if the timer freezes and continues when accidentally logged out and logged in.

EPIC 5 – RESULTS PAGE

Story 1:
Once the quiz gets over the result screen must be shown.

Task 1: UI for result page is designed.
Task 2: Write React code for creating the designed UI.
Task 3: CSS is written to style the page.
Task 4: The button to navigate to the home page is created.
Task 5: Functions for making different API calls are created.
Task 6: Data to be displayed are read from the database.

Story 2:
As a student, I must be able to view my score and correct answers after completing the test.

Task 1: The student response array is compared with the valid response array and the mark is calculated.
Task 2: Calculated marks are updated in the database.
Task 3: Students can give feedback regarding any problems they faced during the quiz.
Task 4: A submission id is generated in the database and displayed on the screen.
Task 5: Once test time gets over, the correct answers are displayed.
Task 6: Along with the mark scored by the student, the rank of that student in that specific test is also calculated and displayed.

Story 3:
The performance of the students must be shown to all stakeholders.

Task 1: Course outcome wise mark or each student is calculated.
Task 2: Mark, percentage, and CO wise mark is updated in the database.
Task 3: A chart showing the performance of that student in each course outcome is displayed.
Task 4: Rank of that student in that test is displayed.
Task 5: A random inspirational quote is displayed.
Task 6: Along with the mark scored, the percentage of the mark is calculated and displayed.

Story 4:
As a faculty, I must be able to analyze the results after conducting a quiz.

Task 1: Class-wise average is calculated.
Task 2: CO-wise class average is calculated.
Task 3: CO-wise class average and the class-wise average is updated in the database.
Task 4: All the averages and list of student marks are displayed to faculty.

Task 5: The faculty can edit the marks obtained by students in case of any discrepancies.
Task 6: An option to download the marks in CSV format is created.

Story 5:
A leader-board must be present honoring the toppers of that specific test.

Task 1: Leader board component UI is designed.
Task 2: React and CSS are done.
Task 3: Mark list of that specific data is read from the database.
Task 4: Mark list is sorted and a rank attribute is added.
Task 5: Mark list with name and mark is displayed in the leaderboard.
Task 6: The leader board is made available on the result page.

Testing Tasks:

Task 1: Check if the page is responsive on different devices.
Task 2: Check if the download as CSV is working correctly.
Task 3: Check if the grading of the quiz is working perfectly.
Task 4: Check if the performance metrics are working error-free.
Task 5: Testing the API with software like postman.
Task 6: Check if the sorting in the leader board is working perfectly.
Task 7: Check if the correct answers option is working fine.
Task 8: Check if the result and quiz details are correctly acquired from the database and shown on the screen.

EPIC 6 – MISCELLANEOUS

Story 1:
Passwords of users are encrypted and stored in the database.

Task 1: Write the encryption function in the backend file.
Task 2: Write the decryption function to authenticate.
Task 3: Add the encryption variable to the database.
Task 4: Configure crypto js.
Task 5: Plan on which encryption.
Task 6: Check if the encryption is viable.

Story 2:
As an admin of the application, I should be able to access and edit everything.

Task 1: Creating admin login credentials.
Task 2: Allowing admin to access all the tables in the database.
Task 3: Creating minimalistic UI for admin with full access.
Task 4: Create an Edit portal for admin.
Task 5: Configure admin push notification.
Task 6: Allow admin to attend every test.

Story 3:
Pushing Notifications.

Task 1: Design the notification component.
Task 2: Create a notification component.
Task 3: Add CSS to design the component.
Task 4: Configure places where notifications are gonna be used.
Task 5: Read the database to get the timing of notifications.
Task 6: Design the exit button.
Task 7: Add a notification to the stack.

Story 4:
As a user, I like to have an option for dark mode.

Task 1: All colours are initialized as variables.
Task 2: Design toggle button UI.
Task 3: Create a toggle button on the home page.
Task 4: Update the status in the database.
Task 5: Read from the database for the toggle status, every time logged in.
Task 6: Animation for change in toggle button.

Story 5:
Developers homage and reporting a bug.

Task 1: Design the developer homage page.
Task 2: Add the developer info.
Task 3: Add a text box to report any bug seen in the app.
Task 4: Setup functions to receive mail.
Task 5: Convert the bug report into a mail and send it to the developer mail.
Task 6: Setup flush bar message once the bug is reported.

Testing Tasks:

Task 1: Check if the password encryption and decryption are happening smoothly.
Task 2: Check if the dark mode toggle doesn't affect any other function.
Task 3: Check if the bug report mechanism works perfectly.
Task 4: Check if the admin can access the database, quizzes, and other functionalities.
Task 5: Testing the API with software like postman.

Development Tasks:

Task 1: Create encryption and decryption functions.
Task 2: Create bug report functionality.
Task 3: Code react functions for pushing notification.
Task 4: Use variables and code to change the value on dark mode toggle.
Task 5: Create Admin page with responsive UI.

General Engineering tasks:

Task 1: Use Invision for visualizing the epics, user stories, and tasks.
Task 2: Create an account in Jira to keep track of the progress of the project.
Task 3: Create MongoDB atlas cluster for the database.
Task 4: Setup AWS for hosting the web app and database in the cloud.
Task 5: Setup Github project repo. Collaborate with the team over the cloud.

MoSCoW Prioritization

Must have:

EPIC 1:

- As a user, I must be able to view the initial Home-Screen which has provisions to enter credentials.
- Comparing input credentials against the identity stored in the backend database.
- As an app-user, I must be redirected to the next page based on my credibility.

EPIC 2:

- As a student, I must be able to see all the courses I registered for and notifications for any quizzes scheduled for the day.
- As a faculty, I should be able to see the list of courses handled by me and ongoing quizzes scheduled by me.

EPIC 3:

- As a Quiz-Setter, I must be able to set-up questions for a quiz.

EPIC 4:

- As a student, I must be able to start the quiz and navigate through the questions.
- As a Quiz taker, I should be able to view the details of each question.

EPIC 5:

- Once the quiz gets over the result screen must be shown.
- As a student, I must be able to view my score and correct answers after completing the test.

Should have:

EPIC 1:

- As an app-user, I must be provided with the provision for Multi-User login: Faculty or Student.

EPIC 2:

- As a faculty, I must be able to schedule a quiz or see all the quizzes conducted in that course.
- As a student, I must be able to see all the quizzes attended in a particular course.

EPIC 3:

- As a faculty, I must be able to choose an appropriate time and duration for the quiz.
- As a student, I must be able to access the quiz at the release time.

EPIC 4:

- As a faculty, I want the performance analysis for each student for the quiz.
- As a Quiz taker, I want to monitor the timer during my course of the test.

EPIC 5:

- The performance of the students must be shown to all stakeholders.
- As a faculty, I must be able to analyze the results after conducting a quiz.

Could have:

EPIC 1:

- Forgot Password Mechanism

EPIC 2:

- Personal information can be viewed and edited in the ABOUT section

EPIC 3:

- As a Quiz-Setter, I can either set/edit the password and release the time of the quiz.

EPIC 4:

- As a faculty, I should have access to the ordering of the questions.

EPIC 5:

- A leader-board must be present honouring the toppers of that specific test.

EPIC 6:

- As an admin of the application, I should be able to access and edit everything.
- Pushing Notifications.

Will not have:

EPIC 3:

- As a faculty, I can manage the feedback of each individual quiz-taker right after they submit.

EPIC 6:

- Passwords of users are encrypted and stored in the database.
- As a user, I like to have an option for dark mode.
- Developers homage and reporting a bug

LEGEND:

T-Shirt Size Standardization:

- <=12 – S (small size)
- >12 & <=17 – M (medium size)
- >=18 – L (large size)

Story Points Standardization:

- Each story point derives into 4 technical working hours
- 1 Story Point – 4 hours
- 2 Story Points – 8 hours
- 3 Story Points – 12 hours …

****