Please carefully read the **README.md** before recording.

PPT Slide 1 (EDFLIX title slide), Niamh

**Introduction**

Hello, the project presented in this design video is the EDFLIX for the client IBM.

PPT Slide 2, Niamh

**Our client**

IBM is a multi-billion-dollar global information technology company which sells both hardware and software to customers around the world. IBM puts huge emphasis on education and self-directed learning, and as such they launched the IBM Skills Build platform in the UK towards the end of 2019, with the goal of supporting learners in developing new skills from secondary level education all the way to entry-level employment. The platform not only offers learning artifacts to those looking to broaden their horizons in any of a large selection of areas within computer science, but also provides hugely valuable resources that are available to download for free at any time.

Unfortunately, the client has had feedback that the currently existing site is difficult to navigate and can be unwieldy due to the large volume of resources available on the platform.

PPT Slide 3, Niamh

**Aims**

We aim to solve this problem by implementing a front-end video streaming style wrapper around the already existing Skills Build site, with an AI supported recommendation system to push learning resources to users based on their interests. We shall also provide a system for rating and commenting learning artifacts for users, supporting the AI to promote content of a higher quality to more of the user-base.

PPT Slide 4, Niamh

**Deliverables**

Our deliverables will be a simple and intuitive web-based user interface allowing learners to easily access any of the learning artifacts or resources they wish. We will create a SQL database with which to store both user data and usage data for each individual learning artifacts. In addition, we plan to create visualisations of the collected data for the client. We will implement our AI for recommendations with IBM Watson, and all code will be stored in a GitHub repository accessible only by our software development team, Durham university, as well as the client, IBM.

PPT Slide 5 (catalogue slide), Niamh

Over the remainder of this presentation, we will give a more detailed technical explanation of both our back and front-end software, including potential features that are in consideration, this will then be followed by a discussion of our development processes, and then our future plans will be detailed.

PPT Slide 6, Lucas

**Technical Outline**

PPT Slide 7, Lucas

**Communication between systems**

Our software contains a front end and a back end. The front end contains everything that can be viewed by the users. The back end contains the database, the artificial intelligence, and the RESTful API. Data and queries, responses of queries and push information are transmitted between the phases of front end and back end.

PPT Slide 8, Lucas

**Data flow**

To begin with, through the sign-up process, the user’s individual information and areas of study is stored in the database, and then used to create a profile webpage for visualisation. When the user updates their profile, the new content will be stored in the database.

PPT Slide 9, Lucas

When the user adds or removes an individual learning artifact to their ‘currently watching’ and ‘to watch’ list, it will be stored in the database together with other personal data of the user, and then sent back to the webpage for visualisation. This data will also be used to train the AI to automatically produce sufficient push information, for example recommended learning artifacts.

PPT Slide 10, Lucas

The list of all existing learning artifacts is also stored in a database and can be visualised by the user through filtering and searching.

PPT Slide 11, Lucas

The detailed information of each existing learning artifact is stored in the database together with its comments, rating, and link to the corresponding IBM course. This data can be viewed by the user through clicking on a learning artifact from other webpages. When the user creates a new comment or a rating, it will also be stored in the database and the web page will be dynamically updated to display this change.

PPT Slide 12, Juntao

**Front end**

For our front end, we will use the tutorLMS framework. This framework allows us to create a learning management system for users, including a sign-in and sign-up page, a profile page for the user to view and edit personal information, an individual learning artifacts page to store user’s selected courses, and an existing learning artifacts page for the user to view, search and filter all currently available courses. The recommended and popular learning artifacts pushed by AI are also presented on the page to give useful advice for the user to select courses. All these pages are linked together by a navigation bar so that the user can easily view different content.

PPT Slide 13, Juntao

The student can click on any learning artifact to view the detailed content of the course. We also provide the functionalities to comments and rating the learning artifact on the page. If the user eventually decides to study the course, they can simply click on the button or URL shown on the page, and then the software will directly lead them to the IMB Study Build course.

PPT Slide 14, Qitao

**Back end – RESTful API**

As an important midpoint of the data transmission, our server will be implemented by PHP and our API will strictly follow the standard of the RESTful API. The RESTful API uses standard HTTP methods, including GET, POST, PUT, PATCH, and DELETE. As presented in the table, all functionalities which interact with the database are assigned to these methods. The RESTful API includes paths pointing to each functionality with clear surface structure and simple parameters. Finally, the RESTful API has the function of monitoring status code and sending the correct one to the front end.

PPT Slide 15, Qitao

**Back end – Database**

Our database will be implemented by MySQL. The database contains six entities, the student entity, the area of study entity, the individual learning artifacts entity, the recommended learning artifacts entity, the existing learning artifacts entity, and the comments entity. Based on the functional requirements, each entity has its related attributes and required fields as shown in the entity-relationship model. In the model, the blue attribute represents the primary key of the entity, and the green attribute represents the foreign key, so that all entities are related together for data querying and data transmission.

PPT Slide 16, Stan

**Back end – AI**

Our artificial intelligence will be implemented by IBM Watson. The AI algorithm learns from the data stored within the student entity, the corresponding individual learning artifacts entity (chosen by the student), and the existing learning artifacts entity. To be specific, the AI algorithm initially utilises the user’s chosen area of study to query a list of relevant and related existing learning artifacts. Next, a push value is calculated for each existing learning artifact according to the user’s year of study and chosen modules, the value of clicks and the difficulties of all individual learning artifacts chosen by the user for current or future learning, and the value of clicks and difficulty of the learning artifact for which the push value is being calculated. Based on the generated push value and its corresponding weight, a finite number of existing learning artifacts will be selected as recommended learning artifacts and appear in the webpage.

PPT Slide 17, Stan

**Potential features – Cookies**

To enhance the utility and convenience of the software, HTTP cookies should be used to save essential user information for future use. For example, in the sign-in process, the fields of email and password will be automatically filled in with the latest content entered by the user. Or, in the function of searching existing learning artifacts, the search bar will provide a list of recently used keywords for the user to choose.

**AI**

To achieve accurate and valid recommendations, we consider building a high-dimensional portrait of the user in the AI, by collecting and analysing more precise user behaviour data, such as the duration of a user's visit to a certain learning artifact and the frequency of a user's search for a certain keyword. For example, if a group of users have similar portraits, and one user among them selects a learning artifact, such learning artifacts should be pushed more frequently to all users in the group.

**Database**

Currently, the learning artifacts in the ‘IBM Skills Build’ site are artificially stored to the database. To increase the convenience of maintenance and decrease the cost, we consider implementing the automatic addition and deletion of the learning artifacts and their corresponding resources through web crawling and advanced database querying.

PPT Slide 18, Ananyaa

**Project Development**

PPT Slide 19, Ananyaa

We’ve decided to use the Dynamic System Development DSDM as our software development framework as we found it best fit our goals and patterns of work. This involves working in pre-planned time boxes, each lasting two weeks, beginning with a team meeting to discuss and assign tasks for each time box. An intermediary meeting within the two weeks is scheduled to allow us to check in with other group members and offer support if anyone feels unable to get the work done in time. After this mid-point, functionality which can be tested should be checked against requirements in time for the group meeting at the end of the timebox, used to review the past two weeks of work and schedule tasks for the next timebox.

PPT Slide 20, Ananyaa

In order to keep track of various tasks given to group members we have decided to use Trello, which allows users to create lists of tasks, move items across lists, and assign priorities.

PPT Slide 21, Ananyaa

**Future plan**

PPT Slide 22, Ananyaa

We will meet weekly to listen to the progress report from each group member and monitor every possible risk or issue that may damage the entire development process.

The programming sub team will continue working on the development of the front end, the database, and the AI. The development of the RESTful API will begin after the first release of these three phases.

The documentation sub team will start to discuss the content of the test plan based on the progress of the programming sub team.

PPT Slide 23, Ananyaa

Before the submission of the test plan, the first release of the software should be completed. This release should contain all core functionalities with a high priority in the requirement specification. Among them, the AI should be implemented to work properly and intelligently. The RESTful API should link all phases in the front end and the back end successfully.

After the submission of the test plan, more releases of the software will be developed based on the functionalities with a medium or low priority and the potential features mentioned before. Eventually, the software will be tested and adjusted to satisfy all non-functional requirements as much as possible before the product handover.

PPT Slide 24, Ananyaa

Thank you for watching.