

# UECS3294 ADVANCED WEB APPLICATION DEVELOPMENT

## PRACTICAL ASSESSMENT PAPER A

Name:

Student ID:

Marks:

/100

**Instruction: This is an open book assessment. Answer all questions. You have 2 hours & 30 minutes to complete this assessment.**

### 1. Managing Application Dependencies, Database Migration & Assets

- (a) Create a new Laravel Project named **papera<your-student-id>**, for example, if your student ID is *1501234*, your project shall be named ***papera1501234***. (5 marks)
- (b) Configure JavaScript scaffolding to use *React* scaffolding. You will need to use the updated **webpack.mix.js** file with Laravel Mix to compile your JavaScript code that you will develop in Q3. (5 marks)
- (c) Using Laravel's database migration, create the following tables:

id	name	created_at	updated_at
1	Parti Ayam	1976-08-19 11:20:12	1976-08-19 11:20:12
2	Parti Itik	1977-07-15 15:30:26	1999-12-31 11:22:42
3	Parti Kucing	1986-06-20 11:30:55	1986-06-20 11:30:55

Table Name: *parties*

id	name	party_id	created_at	updated_at
1	Abu Bakar Muhammad	3	2018-04-21 09:45:01	2018-04-21 09:45:01
2	Ng Pei Li	3	2018-04-21 09:48:03	2018-04-21 09:48:03
3	Ranjit Singh Deo	3	2018-04-21 09:58:02	2018-04-21 09:58:02
4	Foo Yoke Wai	1	2018-04-21 09:58:11	2018-04-21 09:58:11
5	Chia Kim Hooi	2	2018-04-21 09:59:13	2018-04-21 09:59:13

Table Name: *candidates*

- (d) Build an Artisan command to insert three (3) rows of data to *parties* and five (5) rows of data to *candidates*. (5 marks)

### 2. Creating Models, Resources & Controllers

- (a) Create Eloquent model classes for the tables you have created in Q1. Define the necessary *mass-assignable* attributes and *relationship* methods in the model classes. (10 marks)
- (b) Define resources and resource collections for both *Party* and *Candidate* models. Exclude the timestamp attributes and include related data if the relationship is loaded. (15 marks)

- (c) Develop a RESTful API for the resource **candidates** as follows:
- i. **GET /api/candidates**: Returns all candidates including the party's resource in JSON format.
  - ii. **GET /api/candidates/{candidate}**: Returns the candidate including the party's resource specified by the parameter **id** in JSON format.
  - iii. **POST /api/candidates**: Inserts a new candidate record.
  - iv. **PUT /api/candidates/{candidate}**: Updates the candidate record.
- (15 marks)

### 3. Client-side Scripting *React* JavaScript Library

- (a) Develop client-side code using *React* JavaScript library to fetch JSON data from the endpoint **GET /api/candidates** and display the records in HTML Table format. You will need to create a Blade view file to specify a placeholder element for React to render the component. (20 marks)
- (b) Develop client-side code using *React* JavaScript library to fetch JSON data from the endpoint **GET /api/candidates/{candidate}** and display the selected record in any appropriate format. You will need to create a Blade view file to specify a placeholder element for React to render the component. (20 marks)