Engineer to Excel

SAVEETHA
SCHOOL OF ENGINEERING
Affliated to AICTE | IET-UK Accreditation

**PROJECT TITLE**

# ONLINE TICKET BOOKING

**A CAPSTONE PROJECT REPORT**

*Submitted in the partial fulfilment for the award of*

**ITA0239**

**WEB TECHNOLOGY FOR DEVELOPERS**

**Submitted by**

**KANIKA.K(192221083)**

**Under the Supervision of**

**DR.JUDY SUBRAMANIAN**

**NOVEMBER (2024)**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**
**CHENNAI – 602 105.**
**TAMIL NADU, INDIA**

## BONAFIDE CERTIFICATE

This is to certify that the project report entitled **A case study on online booking system** submitted by KANIKA.K –192221083., to Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, is a record of bonafide work carried out by him/her under my guidance. The project fulfils the requirements as per the regulations of this institution and in my appraisal meets the required standards for submission.

DR.JUDY SUBRAMANIAN

COURSE FACULTY
Saveetha School of Engineering,

SIMATS, Chennai - 602105

# ACKNOWLEDGEMENT

This project work would not have been possible without the contribution of many people. It gives me immense pleasure to express my profound gratitude to our Honourable Chancellor **Dr. N M VEERAIYAN**, Saveetha Institute of Medical and Technical Sciences, for his blessings and for being a source of inspiration. I sincerely thank our Director of Academics **Dr. DEEPAK NALLASWAMY,** SIMATS, for his visionary thoughts and support. I am indebted to extend my gratitude to our Director **Dr. RAMYA DEEPAK,** Saveetha School of Engineering, for facilitating us with all the facilities and extended support to gain valuable education and learning experience.

I register my special thanks to **Dr. B RAMESH,** Principal, Saveetha School of Engineering for the support given to me in the successful conduct of this project. I wish to express my sincere gratitude to my Course faculty **Dr.JUDY SUBRAMANIAN**, for his inspiring guidance, personal involvement and constant encouragement during the entire course of this work.

I am grateful to Project Coordinators, Review Panel External and Internal Members and the entire faculty of the Department of Design, for their constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work

# INDEX

## ABSTRACT

Online movie ticket booking system for movies is a web-based program. This application allows users to purchase cinema tickets over the portal. To buy tickets, people must first register or login. This website's backend is PHP and JavaScript, and the front end is HTML and CSS. All phases of the software development life cycle are efficiently managed in order to design and implement software. On the website, there are two panels: one for administrators and one for customers/users. The admin has the ability to add cinemas, movies, delete, halt execution, and add screens, among other things. The website is simple to navigate and appealing, saving the end user time.

## INTRODUCTION

The study's scenario is the online booking system at HOYTS Australia. Customers can utilise the online booking system to make prior reservations, check movie screening times, watch trailers, and read reviews. The business intends to create a web-based gateway for online movie reservations. The online cinema reservation system's goal is to make it possible to book movie tickets over the internet. Customers can view and book programs based on availability. People of all ages can now be seen in movie theatres. People are more concentrated on their work than they have been in the past. People, on the other hand, spend their leisure time watching movies. When people start working, they don't want to waste time waiting in lines to buy movie tickets. We came up with the concept of developing a web-based movie ticketing system because of this. This is a dynamic website where visitors may create a list of movies they want to see, as well as learn about which movies are presented in which rooms, ticket costs, and other relevant info. Consumers (i.e. end users and admin which is responsible for managing data) may immediately feel the benefits of our technology by browsing and using the web protol. This technology is also beneficial to theatre owners since it allows them to easily reach out to clients and grow their business by using an interactive GUI (graphical user interface).

The online movie ticketing system has the following features:

a) User registration and login

b) Simply select the movie, show time, and theatre name

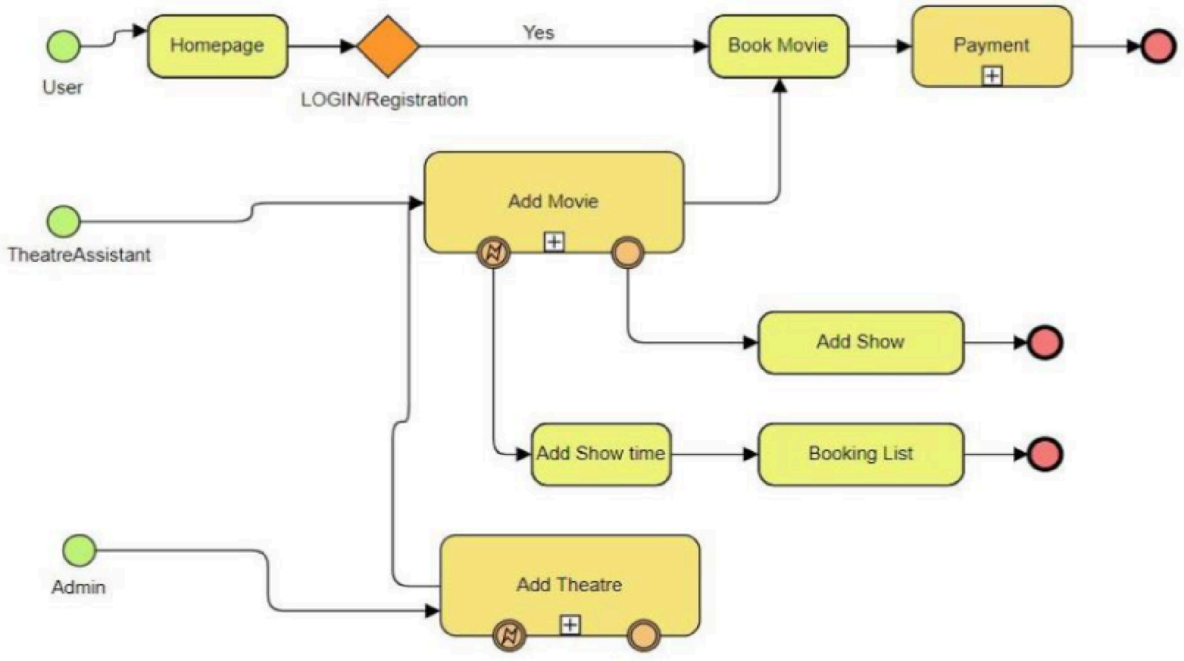
c) Screenings, upcoming movie trailers, and Comments and payment gateway

d) Administrators have access to the dashboard, which allows them to create, delete

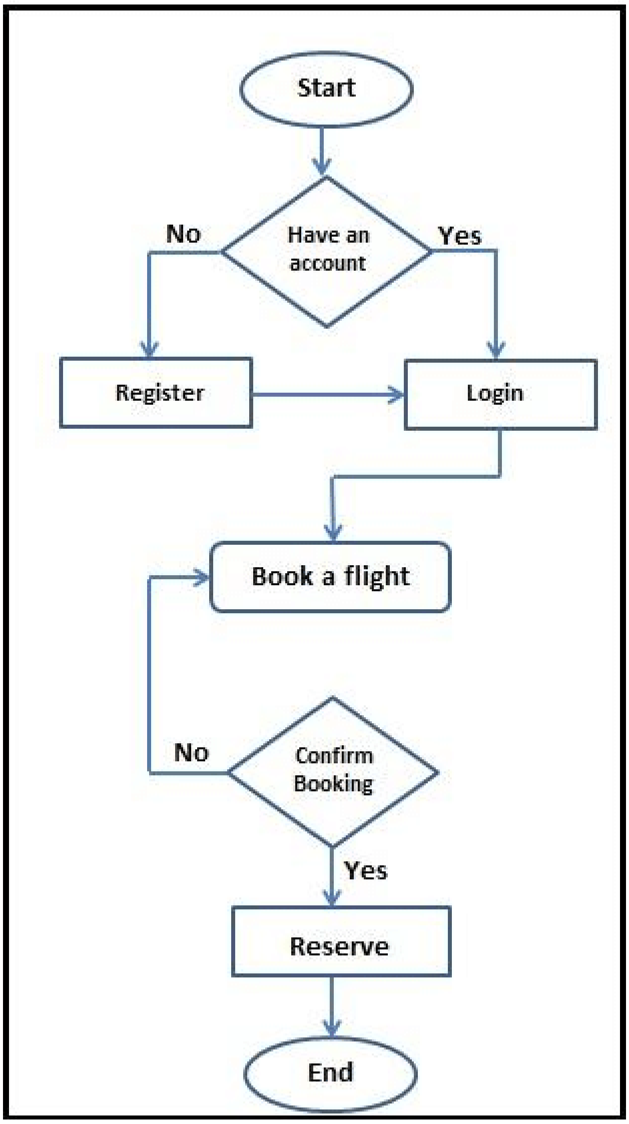
and videos.

## SYSTEM ARCHITECTURE

An architecture is "the fundamental organisation, which comprises the components, relationships between themselves and the surroundings and the rules that govern its development and design. The system architecture process is where concepts are established which constitute the backbone of the current system. It is a model concept which outlines how the planned or an existing system is structured and behaved. The model could comprise technical foundation, user demands and a list of components of the system (hardware and software). During the system architecture process, the major decisions are: The new system's characteristics.The architectural style.Used kind of software (custom or off-the-shelf).Technology types employed.How to deploy the system



## FLOW CHART



**User opens the app**: The user begins by launching the application.

**Selects Event/Travel**: The user navigates to select the event or travel route they want to book a ticket for.

**Check Availability**: The system checks for available tickets.

**Tickets Available**: If tickets are available, the user can select date and seats.

**Tickets Unavailable**: If no tickets are available, the system shows an error or suggests alternative options.

**User Confirms Booking**: The user proceeds to confirm the booking.

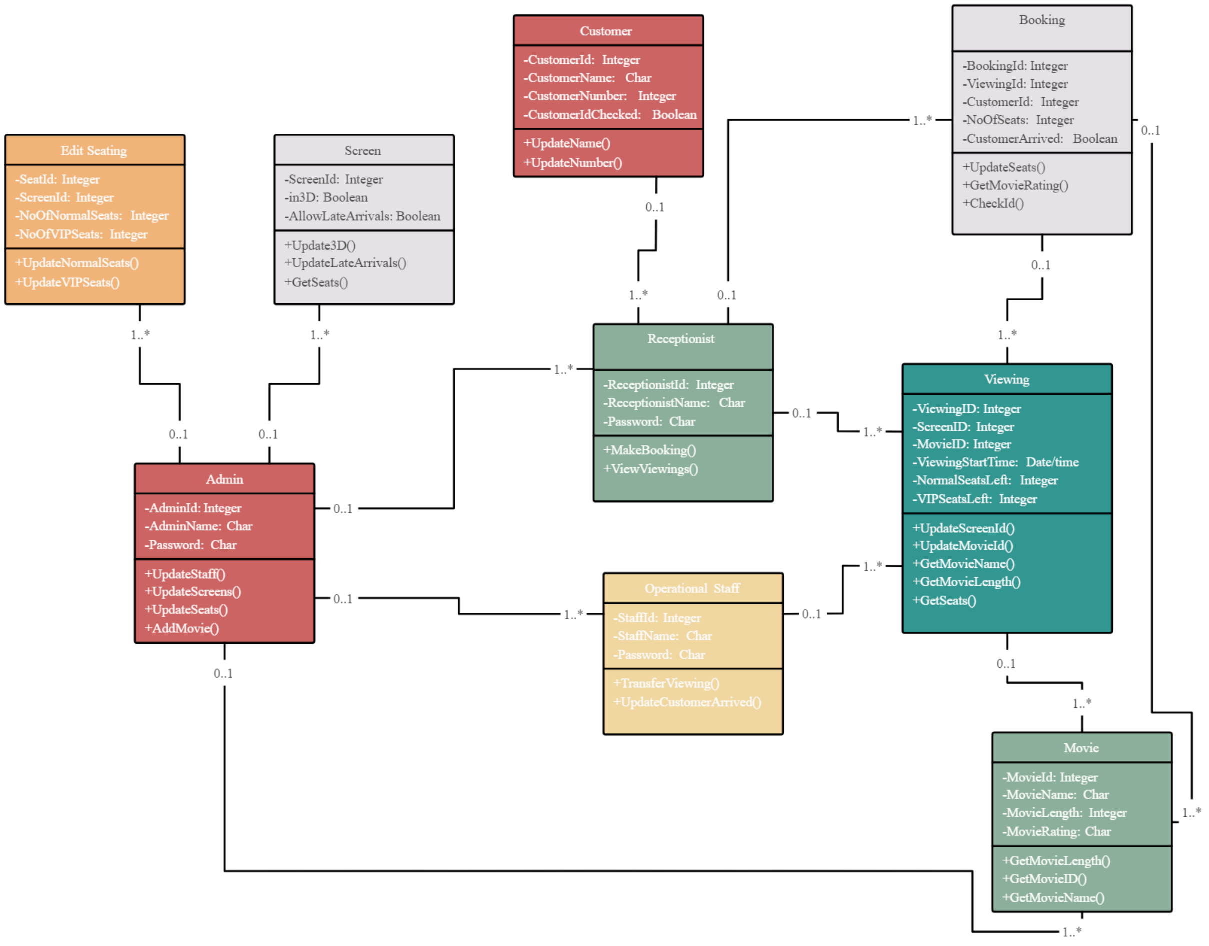
**Payment Processing**: The user enters payment information, and the system processes the payment.

**Payment Success**: Upon successful payment, the system sends a confirmation to the user.

**Payment Failure**: If payment fails, an error message is shown with an option to retry.

**Ticket Generation**: Upon successful booking and payment, a ticket (and potentially a QR code) is generated and sent to the user.

## CLASS DIAGRAM



1. **User**: Represents the customer who uses the app to book tickets.
   - Methods: Search for events, make a booking, view bookings.
2. **Event**: Represents events or travel services offered for ticket booking.
   - Methods: Check availability of tickets, reserve seats.
3. **Booking**: Represents a booking transaction.
   - Methods: Confirm or cancel bookings.

4. **Payment**: Represents the payment process.
    - Methods: Process and refund payments.
5. **Ticket**: Represents the actual ticket issued after a booking.
    - Methods: Generate ticket and QR code for scanning.
6. **Notification**: Represents the notifications sent to users.
    - Methods: Send notifications (e.g., booking confirmation).

**Relationships:**

- **User → Booking**: A user makes one or more bookings.
- **Booking → Payment**: A booking includes a payment process.
- **Booking → Ticket**: A successful booking generates one or more tickets.
- **Booking → Event**: A booking reserves seats for an event.
- **User → Notification**: The user receives notifications for confirmations or updates.

## CODE IMPLEMENTATION

```
public class User {
    private String userId;
    private String name;
    private String email;
    private String phoneNumber;

    public User(String userId, String name, String email, String phoneNumber) {
        this.userId = userId;
        this.name = name;
        this.email = email;
        this.phoneNumber = phoneNumber;
    }

    // Getters and setters
    public String getName() {
        return name;
    }
    public String getEmail() {
        return email;
    }
    public void makeBooking(Event event, int numberOfTickets) {
        Booking booking = new Booking("B001", event, numberOfTickets);
        booking.confirmBooking();
    }
}
public class Booking {
    private String bookingId;
    private Event event;
    private int numberOfTickets;
    private String status;

    public Booking(String bookingId, Event event, int numberOfTickets) {
        this.bookingId = bookingId;
        this.event = event;
        this.numberOfTickets = numberOfTickets;
        this.status = "Pending";
    }

    public void confirmBooking() {
        if (event.checkAvailability(numberOfTickets)) {
            event.reserveSeat(numberOfTickets);
            this.status = "Confirmed";
            System.out.println("Booking confirmed for event: " + event.getEventName());
        } else {
            System.out.println("Booking failed: Not enough available seats.");
        }
    }
}

public class Main {
    public static void main(String[] args) {
```

```java
        User user = new User("U001", "John Doe", "john.doe@example.com", "1234567890");
        Event event = new Event("E001", "Concert", "2024-10-05", "Concert Hall", 100);

        user.searchEvent(event);
        user.makeBooking(event, 2);
    }
}
```

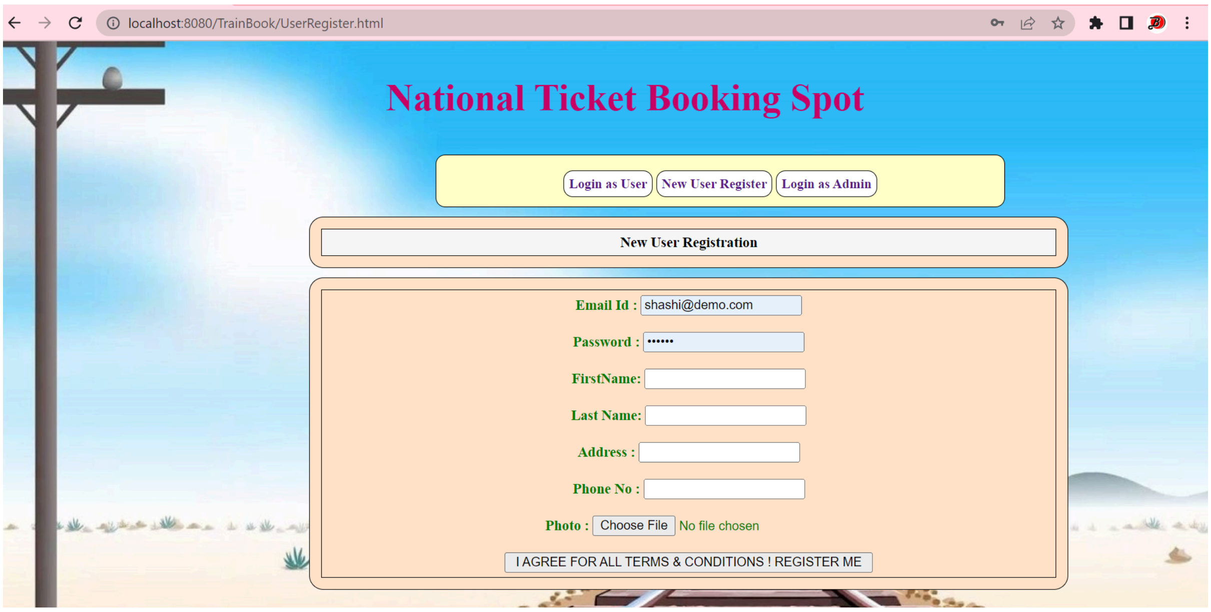
**HTML CODE**

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Ticket Booking App</title>
</head>
<body>
    <h1>Ticket Booking</h1>
    <form action="/booking" method="POST">
        <label for="name">Your Name:</label><br>
        <input type="text" id="name" name="name"><br><br>

        <label for="email">Your Email:</label><br>
        <input type="email" id="email" name="email"><br><br>

        <label for="event">Select Event:</label><br>
        <select id="event" name="event">
            <option value="concert">Concert - 2024-10-05</option>
            <option value="play">Play - 2024-11-15</option>
        </select><br><br>

        <label for="tickets">Number of Tickets:</label><br>
        <input type="number" id="tickets" name="tickets" min="1" max="10"><br><br>
        <input type="submit" value="Book Tickets">
    </form>
</body>
</html>

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/booking")
public class BookingServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        String name = request.getParameter("name");
        String email = request.getParameter("email");
        String eventSelected = request.getParameter("event");
        int numberOfTickets = Integer.parseInt(request.getParameter("tickets"));

         Event event = new Event("E001", eventSelected, "2024-10-05", "Concert Hall", 100); //
Hardcoded for simplicity
        User user = new User("U001", name, email, "1234567890");

        user.makeBooking(event, numberOfTickets);

                response.getWriter().println("Booking confirmed for " + name + " for " +
numberOfTickets + " tickets to " + eventSelected);
    }
}
```

**CSS CODE**

```
/* Global Styles */
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
}

h1 {
    text-align: center;
    color: #333;
    margin-top: 20px;
}

table {
    width: 80%;
    margin: 20px auto;
    border-collapse: collapse;
    background-color: #fff;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

table th, table td {
    padding: 12px;
    border: 1px solid #ddd;
    text-align: center;
}

table th {
    background-color: #007BFF;
    color: white;
}

table tr:nth-child(even) {
    background-color: #f9f9f9;
}

table tr:hover {
    background-color: #f1f1f1;
}

a {
    color: #007BFF;
    text-decoration: none;
}

a:hover {
    text-decoration: underline;
}

/* Form Styles */
form {
    width: 60%;
    margin: 20px auto;
    background-color: #fff;
    padding: 20px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    border-radius: 5px;
}

form label {
    display: block;
    margin-bottom: 8px;
    font-weight: bold;
    color: #333;
}

form input[type="text"], form input[type="email"], form input[type="number"] {
    width: 100%;
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ddd;
    border-radius: 4px;
}

form button {
    width: 100%;
    padding: 12px;
    background-color: #28a745;
    color: white;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    font-size: 16px;
}

form button:hover {
    background-color: #218838;
}

/* Responsive Design */
@media (max-width: 768px) {
    table {
        width: 95%;
    }

    form {
        width: 90%;
    }

    form button {
        font-size: 14px;
    }
}
```

## OUTPUT


localhost:8080/TrainBook/UserRegister.html
Login as User
New User Register
Login as Admin
Choose File
I AGREE FOR ALL TERMS & CONDITIONS ! REGISTER ME

National Ticket Booking Spot

New User Registration

Email Id : shashi@demo.com

Password : ••••••

FirstName:

Last Name:

Address :

Phone No :

Photo : No file chosen

## CONCLUSION

The **Online Ticket Booking System** effectively demonstrates a seamless integration between backend and frontend technologies to provide a functional and user-friendly platform for event ticket reservations.This project serves as a strong foundation for building scalable ticketing platforms. It can easily be extended to include features like **user authentication**, **payment integration**, **seat selection**, and more advanced **analytics** for tracking bookings and revenue.

# REFERENCES

1.Acharya, Kamal, Online Job Portal Management System (May 5, 2024). Available at SSRN: https://ssrn.com/abstract=4817534 or http://dx.doi.org/10.2139/ssrn.4817534

2.Acharya, Kamal, Employee leave management system. (May 7, 2024). Available at SSRN: https://ssrn.com/abstract=4819626 or http://dx.doi.org/10.2139/ssrn.4819626

3.Acharya, Kamal, Online electricity billing project report. (May 7, 2024). Available at SSRN: https://ssrn.com/abstract=4819630 or http://dx.doi.org/10.2139/ssrn.4819630

4.Acharya, Kamal, POLICY MANAGEMENT SYSTEM PROJECT REPORT. (December 10, 2023). Available at SSRN: https://ssrn.com/abstract=4831694 or http://dx.doi.org/10.2139/ssrn.4831694

5.Acharya, Kamal, Online job placement system project report. (January 10, 2023). Available at SSRN: https://ssrn.com/abstract=4831638 or http://dx.doi.org/10.2139/ssrn.4831638

6.Acharya, Kamal, Software testing for project report. (May 16, 2023). Available at SSRN: https://ssrn.com/abstract=4831028 or http://dx.doi.org/10.2139/ssrn.4831028

7.Acharya, Kamal, ONLINE CRIME REPORTING SYSTEM PROJECT. (August 10, 2022). Available at SSRN: https://ssrn.com/abstract=4831015 or http://dx.doi.org/10.2139/ssrn.4831015

8.Acharya, Kamal, Burber ordering system project report. (October 10, 2022). Available at SSRN: https://ssrn.com/abstract=4832704 or http://dx.doi.org/10.2139/ssrn.4832704

9.Acharya, Kamal, Teachers Record Management System Project Report (December 10, 2023). Available at SSRN: https://ssrn.com/abstract=4833821 or http://dx.doi.org/10.2139/ssrn.4833821

10.Acharya, Kamal, Dairy Management System Project Report (December 20, 2020). Available at SSRN: https://ssrn.com/abstract=4835231 or http://dx.doi.org/10.2139/ssrn.4835231